

**ESCOLA POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA**

INGENIERÍA EN INFORMÁTICA

**Modelo de cuentas estelares
de la Vía Láctea**

Autor: Jordi Molgó Sendra

Directores: Josep M. Miret

Ramiro Moreno

*Dedicado a Merche y Oriol, las dos estrellas
más brillantes de mi rincón en la galaxia*

Índice general

1. Introducción	4
1.1. Motivación	4
1.2. Introducción histórica a los modelos de cuentas estelares y al conocimiento de la morfología de la Vía Láctea	5
1.3. Estructura del documento	10
2. Preliminares astrofísicos	12
2.1. Sistemas de coordenadas	12
2.2. Sistema de magnitudes	15
2.3. Estructura a gran escala de la Vía Láctea	16
3. Modelo de cuentas estelares de la Vía Láctea	21
3.1. Conversión a coordenadas galactocéntricas	22
3.2. Extinción	23
3.3. Función de luminosidad	25
3.4. Funciones de densidad	27
3.4.1. Disco	27
3.4.2. Bulbo	31
3.4.3. Barra	33
3.4.4. Brazos espirales	34
3.4.5. Halo	36
3.5. Ajustes de los parámetros estructurales	37
4. Implementación	39
4.1. Gestión del proyecto	39
4.1.1. Planificación	39
4.1.2. Gestión de la base de código y control de versiones . .	41
4.2. Consideraciones de diseño	41

4.2.1.	Lenguaje de programación	42
4.2.2.	Patrones GRASP	43
4.2.3.	Patrones GOF	44
4.3.	Arquitectura de la aplicación	45
4.3.1.	Modelo 4+1	45
4.3.2.	Casos de uso	47
4.3.3.	Vista de desarrollo	53
4.3.4.	Vista lógica	54
4.3.5.	Vista de proceso	63
4.3.6.	Vista física	64
4.4.	Detalles de implementación	68
4.4.1.	Implementación de la integración de la función de luminosidad	68
4.4.2.	Implementación de las funciones de densidad	71
4.4.3.	Inicialización retardada	74
4.4.4.	Implementación de la clase <i>ModelParameters</i>	75
4.4.5.	Implementación de la selección en el catálogo	76
4.5.	Paralelización	77
4.5.1.	SMP	77
4.5.2.	MPI	80
4.6.	Uso como librería	82
4.7.	Interfaz web	83
4.8.	Aplicaciones de usuario	83
5.	Resultados	86
5.1.	Datos observacionales	87
5.1.1.	2MASS	87
5.1.2.	Sloan Digital Sky Survey	88
5.2.	Ajuste de los parámetros del modelo a los datos observacionales	88
5.2.1.	Búsqueda de regiones para el ajuste de cada componente	89
5.2.2.	Resultado del ajuste a los datos observacionales	93
5.3.	Comparación multibanda con datos observacionales	101
6.	Conclusión	105
6.1.	Trabajos futuros	106
7.	Bibliografía	108

Capítulo 1

Introducción

En la siguiente memoria se presenta la implementación de un código informático de un modelo (al que hemos llamado Galiac) de cuentas estelares de la Vía Láctea, realizado en el contexto de la asignatura de “Trabajo de Fin de Carrera” de la Ingeniería Informática en la Universidad de Lleida y en colaboración con investigadores del Instituto de Astrofísica de Canarias (IAC).

1.1. Motivación

Los modelos de cuentas estelares no son un concepto nuevo en el campo de la investigación sobre la estructura de nuestra galaxia. Sin embargo, hay una serie de factores que han contribuido a despertar el interés del grupo de investigación del IAC de morfología y dinámica de la Vía Láctea en la creación del modelo que se presenta en este Trabajo de Fin de Carrera.

Este grupo de investigación ha realizado numerosos trabajos acerca de la estructura galáctica, entre los cuales son de especial relevancia los estudios sobre la estructura del disco fino (con sus varias sub-componentes), el bulbo y la barra larga, que han sido publicados en diversos artículos a lo largo de la última década (por ejemplo, López-Corredoira et al. (2002) para el disco fino, López-Corredoira et al. (2005) para el bulbo, o López-Corredoira et al. (2007) para la barra). Sin embargo, el grupo carece de un código propio que unifique en un solo modelo común el conocimiento actual sobre la estructura galáctica. Además, hay otros componentes estructurales de la Vía Láctea que no han recibido tanta atención por parte del grupo, como el disco grueso, el

halo, los brazos espirales o la extinción, y para los cuales ha sido necesario actualizarse a los últimos resultados de la literatura disponible. A raíz de estos motivos ha surgido el interés del grupo en realizar una puesta al día sobre los conocimientos de estructura galáctica, y en particular la construcción de este modelo unificado y actualizado que pueda servir de base a estudios futuros.

Por otra parte, otro factor que ha contribuido a despertar el interés en disponer de un modelo propio es que la mayor parte de los modelos de referencia disponibles hoy en día presentan alguna característica que no se adapta bien a las necesidades de los investigadores del grupo. En su mayor parte estos inconvenientes son por dificultades en su uso, particularmente a la hora de realizar cálculos con conjuntos de datos grandes, o por no hallarse actualizados a los resultados recientes sobre la estructura galáctica. Así, por ejemplo, uno de los trabajos de referencia fue el modelo SKY (Wainscoat et al. 1992), pero éste no incorpora la barra larga, el alabeo y ensanchamiento del disco, y las funciones de luminosidad que usa para algunos componentes se ha visto con posterioridad que no se ajustan bien a las observaciones; el modelo de Besançon (Robin et al. 2003), es uno de los grandes referentes hoy en día, pero que impone su uso exclusivamente a través de una interfaz *web*, y carece de flexibilidad para adaptarse a necesidades como las de realizar cálculos de regiones extensas del cielo o usarlo para ajustar datos observacionales; etc.

A parte de la construcción del modelo en sí, el segundo objetivo es realizar ajustes a los últimos datos observacionales, que aporten nueva información. Por lo tanto, uno de los requerimientos del modelo es que debe ser eficiente en tiempo de cálculo, a fin de que sea posible usarlo con conjuntos extensos de datos de entrada.

Finalmente, el modelo de cuentas estelares sería un primer paso para otro proyecto más ambicioso: un modelo autoconsistente de la Galaxia, que tenga en cuenta la morfología, la cinemática, abundancias químicas y otros parámetros estelares.

1.2. Introducción histórica a los modelos de cuentas estelares y al conocimiento de la morfología de la Vía Láctea

La franja de luz que se puede observar resaltando sobre el cielo de una noche oscura (hoy en día, por desgracia, ya sólo al alcance de unos pocos) ha

sido, sin duda, una de las visiones que han impactado el imaginario colectivo de la humanidad: ciertamente la mayoría de las culturas humanas que han existido o siguen existiendo en la Tierra han desarrollado alguna explicación mitológica para darle sentido. Sin embargo, aunque el interés de los científicos y filósofos clásicos por la cosmología (aquel perfecto mundo supra-lunar) data de fechas muy antiguas, no se desarrollaron intentos de explicaciones convincentes sobre la naturaleza de esta franja misteriosa hasta tiempos mucho más cercanos.

El primer científico occidental que vislumbró la verdadera esencia de la Vía Láctea fue Galileo, cuando en 1610 apuntó su rudimentario telescopio hacia ella. Se dio cuenta de que aquello que parecía un fluido luminoso era, en realidad, una miríada de pequeñas estrellas que el ojo humano no consigue resolver. La naturaleza estelar de aquella franja se revelaba ante el conocimiento occidental, aunque en aquel momento ni Galileo ni el resto de científicos de la época se dieron cuenta de un aspecto fundamental: el hecho que la Vía Láctea es en realidad una sobre-densidad de estrellas que sigue una línea implica que la distribución de éstas en el cielo no es homogénea ni isotrópica, y por lo tanto no es compatible con una distribución esférica.

El siguiente gran paso hacia el conocimiento de la naturaleza de la Vía Láctea lo dieron Thomas Wright e Immanuel Kant, en 1750 y 1755 respectivamente. El primero sugirió que la apariencia de la Vía Láctea era debido a un efecto óptico causado por el hecho que el Sol esté inmerso en una distribución más o menos aplanada de estrellas. Kant desarrolló esta idea de Wright, argumentando que la acción de la fuerza de la gravedad entre las estrellas tiene que producir una disposición parecida a la del sistema solar: una atracción hacia un centro común que es contrarrestada por el movimiento de rotación de los elementos de masa (los planetas en el sistema solar, las estrellas en la Galaxia), lo cual se termina reflejando en una estructura con forma de disco. Esta teoría, ciertamente, tiene un parecido bastante elevado con nuestra visión actual de la Galaxia. Además, Kant también defendió la hipótesis (antes formulada por Wright) de que las nebulosas podían ser otros sistemas estelares como el nuestro (a los que llamó *universos isla*), pero situados a distancias tan grandes que no era posible resolver sus estrellas.

A finales del siglo XVIII, a medida que las capacidades de los telescopios iban en aumento, fueron apareciendo catálogos estelares y de nebulosas, y mejorando los conocimientos acerca de estas últimas. Uno de los primeros catálogos de nebulosas fue el de Messier, que se planteó inicialmente como una medida para no confundir las nebulosas con los objetos que despertaban

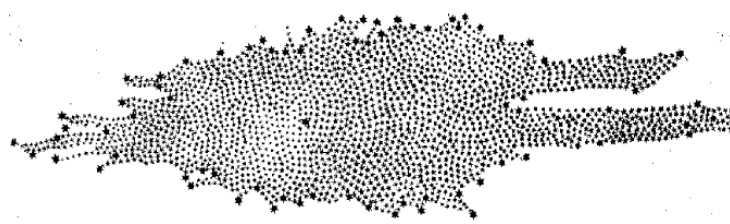


Figura 1.1: Modelo galáctico de Herschel

el interés de la época: los cometas. Pero el catálogo de Messier, de 110 objetos (103 en su publicación original) fue rápidamente superado por el catálogo que compiló el astrónomo inglés William Herschel, de unas 5000 nebulosas. Herschel pudo resolver en estrellas algunas de estas nebulosas, lo cual reforzaría la hipótesis de la existencia de sistemas estelares independientes del nuestro. Además, Herschel fue el primero que desarrolló un modelo sobre la estructura de nuestra galaxia en base a observaciones cuantitativas. Para ello desarrolló la técnica que llamó “conteo de estrellas” (*star gauging*), en la cual contó el número de estrellas que podía observar hasta cierta magnitud aparente en varias zonas del cielo. Posteriormente, mediante las hipótesis (incorrectas) de que las estrellas tienen aproximadamente la misma luminosidad intrínseca y que se hallan repartidas de una forma más o menos uniforme en el espacio, y que se podía observar hasta el borde de la galaxia, supuso que el número de estrellas que se observan en una cierta dirección depende solamente de la profundidad hasta la que se extiende el estrato estelar. El resultado fue el modelo de la figura 1.1, en el cual el Sol reside aproximadamente en el centro de un sistema estelar aplanado y de una forma irregular, cuya longitud es unas 5 veces mayor que su altura. Herschel no tenía ninguna medida de la luminosidad de las estrellas, con lo cual tampoco podía estimar las distancias a ellas y poner valores absolutos al tamaño de su modelo de galaxia.

A finales del siglo XIX las técnicas observacionales cambiaron con la aparición de las placas fotográficas. Esto supuso un gran avance, puesto que permitían integrar la luz captada por el telescopio durante periodos de tiempo largos con lo cual era posible observar objetos mucho más débiles, y además las observaciones quedaban grabadas y, por lo tanto, era posible realizar medidas mucho más rigurosas que las que se pueden tomar con un ocular. Sin embargo, aún no había pruebas concluyentes sobre la escala de nuestra

galaxia, y sobre la naturaleza galáctica o extra-galáctica de las nebulosas.

El primer intento moderno de medir las dimensiones absolutas de la Vía Láctea vino de parte del astrónomo holandés Jacobus Kapteyn en 1906, el cual inició un ambicioso y extenso plan de observación que involucró varios observatorios repartidos por todo el mundo, para obtener placas fotográficas de 200 áreas del cielo. Cada placa fotográfica se estudió minuciosamente, realizando el recuento de estrellas, así como medidas de la luminosidad de cada una y de las variaciones de posición de las estrellas en las diferentes placas de cada campo para determinar su movimiento propio. Así pues, su método era en esencia el de Herschel, pero su aplicación difería en tres puntos importantes: se disponía de una serie de datos observacionales mucho más extensa y profunda que la de Herschel, se sabía que las estrellas tienen luminosidades intrínsecas diferentes, y además conocía la distancia hasta algunas de ellas. El resultado, sin embargo, fue un modelo parecido al que había obtenido Herschel: un esferoide achatado, con el Sol próximo a su centro. Las dimensiones absolutas que obtuvo fueron de 15 kpc para el semieje mayor, y 3 Kpc para el semieje menor. La posición céntrica del Sol en el Universo de Kapteyn, al igual que en el de Herschel, está causada por ignorar los efectos de la extinción del medio interestelar. Herschel no conocía este efecto, pero Kapteyn sí era consciente de su posible existencia. El proceso físico conocido en la época que podía ser la causa de una hipotética extinción en el medio era la dispersión (*"scattering"*) Rayleigh, que es más sensible a longitudes de onda azules que a las rojas. Por lo tanto, Kapteyn realizó una serie de medidas del enrojecimiento relativo entre las medidas fotométricas de placas fotográficas azules y las obtenidas visualmente, y el resultado fue que el enrojecimiento medido era muy bajo. Hoy en día sabemos que la extinción está más causada por la absorción por parte del polvo que por la dispersión por parte del gas, pero este hecho era ignorado en la época y es el que causó que, si bien los métodos de Kapteyn eran correctos, los resultados que obtuvo distan mucho de la morfología actualmente aceptada.

Durante la primera mitad del siglo XX se fue haciendo cada vez más evidente que el modelo de Kapteyn no podía ser correcto. En 1918 el astrónomo Harlow Shapley presentó un modelo de galaxia radicalmente diferente al de Kapteyn, y que obtuvo mediante un acercamiento nuevo al problema: el estudio de la distribución de los cúmulos globulares. A diferencia de las estrellas, que aparecen concentradas en el plano, los cúmulos globulares están distribuidos por todo el cielo. Sin embargo, Shapley observó que esta distribución no es isótropa en longitud galáctica, sino que está más concentrada hacia la

región de Sagitario. Shapley argumentó que los cúmulos deberían estar distribuidos uniformemente alrededor del centro galáctico, y que por lo tanto la asimetría observada indica que el Sol se encuentra a unos 15 kpc del centro. Usando medidas de distancia obtenidas a partir de estrellas variables, estimó que el tamaño del sistema de cúmulos globulares era de unos 100 kpc. Adicionalmente, esta estimación de la escala de la galaxia, que ampliaba enormemente el tamaño que hasta la época se había supuesto para el Universo, le hizo dudar de la naturaleza extra-galáctica de las nebulosas. Sin embargo, las medidas de Shapley estaban sobre-estimadas a causa de un error sistemático en la relación periodo-luminosidad estimada para las cefeidas de los cúmulos globulares. Paralelamente, en la misma época surgió otra corriente de opinión liderada por Heber Curtis, que culminó en el llamado “Gran Debate”. Según esta visión alternativa, la morfología de la galaxia sí se correspondía aproximadamente con el modelo de Kapteyn, y poniendo especial énfasis en la demostración de la naturaleza extragaláctica de las nebulosas (con lo cual se validaría la hipótesis de los “universos isla”, es decir, que algunos de los objetos nebulares observados son en realidad otras galaxias como la nuestra). Sus argumentos sobre la morfología de nuestra galaxia se basaban en el procedimiento más clásico del conteo de estrellas, y las hipótesis sobre la naturaleza extragaláctica de las nebulosas se fundamentaba en observaciones de novae (que en realidad eran supernovas) y en las medidas de los desplazamientos al rojo de estas nebulosas, que sugieren movimientos propios que serían demasiado elevados si se trataran de objetos locales. Finalmente, no se puede afirmar que hubiera ningún ganador en el Gran Debate, puesto que ambos grupos acertaron en algunos de sus argumentos, pero fallaron en los otros: así, Shapley estaba en lo correcto respecto a la posición del Sol en la galaxia, y parcialmente respecto al tamaño de la misma (aunque sus medidas estaban sobre-estimadas), pero falló al considerar que las nebulosas eran de naturaleza galáctica; en cambio, Curtis falló en su modelo de galaxia, pero acertó en la distancia hacia las nebulosas, que las sacaba fuera del contexto de nuestra galaxia y eran, por lo tanto, otras galaxias como la nuestra. El debate se zanjó cuando Hubble descubrió la relación entre desplazamiento al rojo y distancia, y cuando Baade corrigió la relación periodo-luminosidad de las cefeidas de los cúmulos globulares, dando al modelo de Shapley unas dimensiones correctas.

Posteriormente, en la segunda mitad del siglo XX se realizaron avances vertiginosos en nuestro conocimiento sobre la morfología galáctica. En 1944 H. C. van de Hulst predijo la emisión del hidrógeno neutro en la línea de

21 cm. Este hecho ha supuesto un gran avance observacional puesto que el medio interestelar es prácticamente transparente a la radiación de esta longitud de onda, y por lo tanto permite detectar y compilar mapas de la distribución del gas en toda la galaxia mediante medidas cinemáticas. Estos mapas mostraron claramente la naturaleza espiral de la Vía Láctea, aunque debido a varias causas (por ejemplo, movimientos no circulares locales) no es posible extraer de ellos información precisa sobre la morfología exacta de los brazos.

Otro gran paso en el conocimiento del contenido estelar de la galaxia se dio cuando surgieron las técnicas observacionales en el infrarrojo. Estas longitudes de onda están menos afectadas por la extinción, con lo cual es posible obtener información de zonas más alejadas de la galaxia, y sin el “ruido” causado por la extinción (que es de naturaleza grumosa). Así, a finales del siglo XX fue la época de los cartografiados en infrarrojo, como el TMSS (Neugebauer and Leighton 1969), el TMGS (Garzón et al. 1993), DENIS, IRAS, el mapeado de la emisión difusa en infrarrojo por parte del instrumento DIRBE del satélite COBE, o posteriormente el cartografiado del *Two Micron All Sky Survey* (Skrutskie et al. 2006).

Paralelamente, fueron surgiendo varios modelos que intentan explicar las cuentas estelares observadas, y ajustar los resultados a los cartografiados existentes. Uno de los trabajos pioneros fue el de Bahcall and Soneira (1980), que incluía el disco y el esferoide. Posteriormente, fueron apareciendo el modelo SKY (Wainscoat et al. 1992), el de Besançon (Robin et al. 2003), el TRILEGAL (Girardi et al. 2005), etc.

1.3. Estructura del documento

El presente documento está dividido en los siguientes capítulos:

- El primer capítulo, con la introducción histórica al conocimiento de la estructura galáctica y con las motivaciones que han sido el referente para la elaboración de un modelo nuevo de cuentas estelares.
- El segundo capítulo realiza una pequeña introducción a la estructura galáctica y otros conceptos físicos relacionados
- El tercer capítulo contiene la descripción de los detalles de los componentes del modelo que hemos implementado

- El cuarto capítulo hace referencia a cuestiones de implementación, que han sido relevantes a la hora de cumplir con los requisitos y satisfacer las necesidades del modelo
- El quinto capítulo contiene los resultados de las cuentas estelares que se obtienen del modelo, así como su comparación y ajuste con datos observacionales
- Finalmente, el sexto capítulo contiene las conclusiones del trabajo, así como las futuras ampliaciones previstas

Capítulo 2

Preliminares astrofísicos

En este capítulo se va a realizar una pequeña introducción a algunos conceptos astrofísicos que posteriormente se usarán a lo largo del documento.

2.1. Sistemas de coordenadas

Para la implementación del modelo galáctico hemos usado varios sistemas de coordenadas: coordenadas galácticas, galactocéntricas, eclípticas y ecuatoriales. De estos sistemas, los más relevantes para el estudio de la Galaxia son los dos primeros.

Las **coordenadas galácticas**, o coordenadas galácticas heliocéntricas, son un sistema de coordenadas esféricas cuyo origen se sitúa en el Sol. En la figura 2.1a se muestra un gráfico que ilustra su definición. Al tratarse de coordenadas esféricas, tienen dos componentes: la longitud galáctica (que se suele denominar con el símbolo l) y la latitud galáctica (que se denomina con el símbolo b). Como se ha dicho, el Sol es el centro de la esfera en que están definidas, y el plano fundamental del sistema de coordenadas es el plano galáctico. La latitud galáctica es el ángulo que forma el objeto de interés con el plano galáctico y el Sol, con el criterio de signos positivo hacia el polo Norte galáctico y negativo hacia el polo Sur. Por otra parte, la longitud galáctica se mide a partir de la línea que une el Sol con el centro galáctico, y es positiva hacia la dirección marcada por la rotación del Sol alrededor de la galaxia. En ocasiones se complementa el par (l, b) de las coordenadas galácticas con la distancia desde el Sol hasta el objeto (el radio heliocéntrico r), con lo cual entonces ya no se pueden considerar como coordenadas esféricas sino

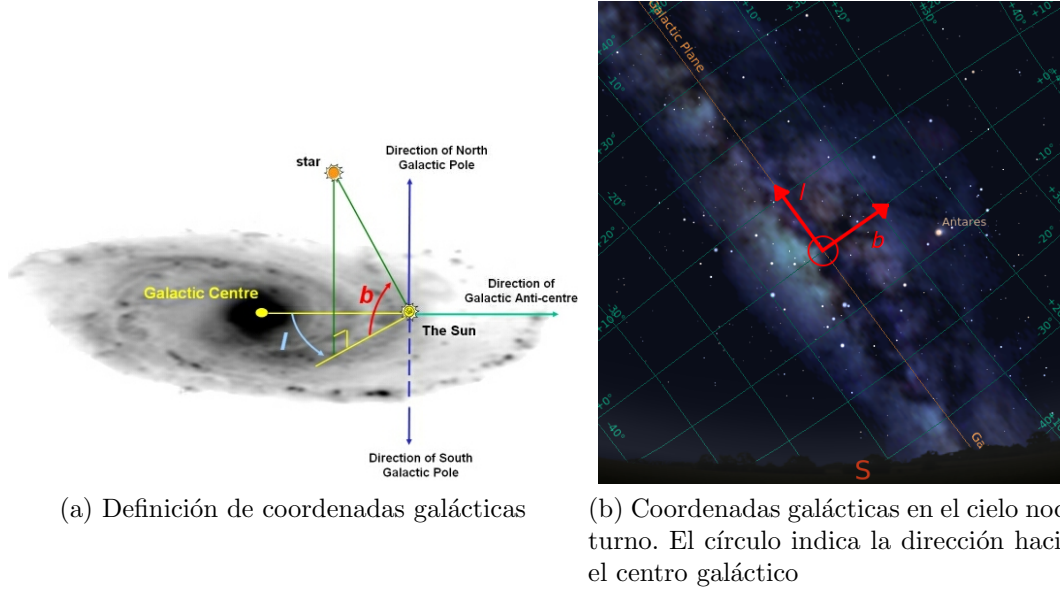


Figura 2.1: Coordenadas galácticas

espaciales. Como se puede ver en la figura 2.1b, las coordenadas galácticas tienen una motivación principalmente observacional.

Las **coordenadas galactocéntricas** (ver figura 2.2), en cambio, tienen su fundamentación en la propia estructura de la Galaxia. Son coordenadas espaciales, y su origen se sitúa en el centro galáctico. Como el componente principal de una galaxia espiral es el disco, al igual que en las coordenadas galácticas se define como plano fundamental del sistema el plano galáctico. Normalmente se trabaja con coordenadas cartesianas galactocéntricas, o con coordenadas cilíndricas galactocéntricas. Para las coordenadas cartesianas galactocéntricas, se usa la línea que une el Sol con el centro galáctico como el eje x , siendo positivo en el sentido del centro galáctico hacia el Sol; el eje y está situado en el plano y perpendicular al eje x , y cuyo sentido está definido de forma que la componente y de la velocidad del Sol tiene signo positivo; finalmente, el eje z es perpendicular a los otros dos (y por lo tanto, perpendicular al plano), con sentido positivo hacia el polo Norte galáctico. Por otra parte, las coordenadas cilíndricas galactocéntricas están compuestas por el radio galactocéntrico R , que es la distancia desde el punto de interés hasta el centro galáctico, el azimuth galáctico θ , que es el ángulo formado entre el

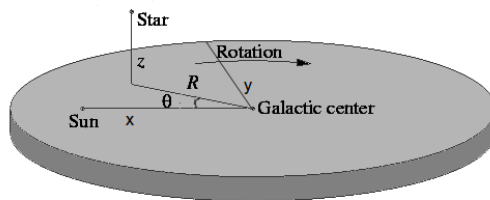


Figura 2.2: Coordenadas galactocéntricas

eje x y la línea que une el punto con el centro galáctico, y la altura sobre el plano z , que es igual que el eje z de las coordenadas cartesianas.

Las **coordenadas eclípticas** se usan sobretodo para la representación de órbitas en el Sistema Solar. Su plano fundamental es la eclíptica (el plano definido por la traslación de la Tierra alrededor del Sol). Su origen puede ser la Tierra o el Sol, con lo cual realmente debemos hablar de coordenadas eclípticas heliocéntricas o coordenadas eclípticas geocéntricas. La latitud eclíptica (b en caso de coordenadas heliocéntricas, y β en caso de ser geocéntricas) es el ángulo entre el plano de la eclíptica y el punto en cuestión, siendo positivo en la dirección del polo Norte terrestre, y la longitud eclíptica (l para las coordenadas eclípticas heliocéntricas y λ para las geocéntricas) es el ángulo entre el punto dado y el punto donde se cruzan el plano de la eclíptica con el plano del ecuador terrestre. En este proyecto, las coordenadas eclípticas se han usado en la consulta de los datos FIR de DIRBE/COBE (en el cálculo de la extinción), puesto que están dados en este sistema de referencia.

Finalmente, las **coordenadas ecuatoriales** son las más utilizadas en el ámbito de la astrofísica en general (aunque no en el ámbito del estudio de la estructura de la Vía Láctea). Las coordenadas ecuatoriales son coordenadas esféricas, y también tienen un origen primordialmente observacional. El origen del sistema se sitúa en el centro de la Tierra, y el plano fundamental es el ecuador terrestre. La coordenada que sería el análogo a la latitud se denomina *declinación* (δ), y es la distancia angular desde la proyección en el cielo del ecuador terrestre hasta el objeto celeste de interés, con sentido positivo hacia el Norte celeste. La coordenada análoga a la longitud se llama *ascensión recta* (α), y es la distancia angular medida desde el punto de in-

tersección entre la proyección celeste del ecuador con el plano de la eclíptica. A diferencia de otras coordenadas angulares, la ascensión recta no se suele medir en grados sino en horas, minutos y segundos de tiempo. Esta particularidad es por conveniencia observacional, y ligada a la rotación de la Tierra: la ascensión recta de los puntos que cruzan en cada momento el meridiano local (el círculo maximal que pasa por el polo celeste y el zenith) es igual a la hora sidérea. En el presente proyecto las coordenadas ecuatoriales se han usado simplemente por el motivo que la mayor parte de catálogos están dados con este sistema, pero inmediatamente las hemos convertido a coordenadas galácticas y hemos trabajado posteriormente con éstas últimas.

2.2. Sistema de magnitudes

Las magnitudes astronómicas son una medida logarítmica y relativa del brillo de un objeto astronómico. Si F_0 es el flujo de luz que recibimos de un objeto, y F_1 es el flujo de otro objeto, la diferencia de magnitudes entre ambos se define como

$$m_1 - m_0 = -2,5 \log 10 \left(\frac{F_1}{F_0} \right) \quad (2.1)$$

Como se puede ver, un aumento de magnitudes realmente significa una disminución en el brillo.

A pesar de que las magnitudes se definen de forma relativa, todos los sistemas fotométricos definen el flujo asociado a una magnitud 0 en cierta banda/filtro y, por lo tanto, es común decir que cierto objeto tiene determinada magnitud en un filtro dado. Por ejemplo, el sistema fotométrico Johnson define que el flujo que recibimos de Vega es el correspondiente a la magnitud 0 en todos sus filtros.

Por otra parte, las magnitudes pueden ser aparentes o absolutas. Un objeto astronómico tiene cierta luminosidad, que es intrínseca a sus propiedades físicas. Sin embargo, a medida que la luz se propaga desde el objeto se va atenuando con una ley de cuadrado inverso. Una magnitud aparente (que se suele simbolizar con m) se calcula con el flujo que recibimos del objeto en la Tierra, y por lo tanto tiene incluido el efecto de la distancia, entre otros. Por el contrario, las magnitudes absolutas están relacionadas con la luminosidad intrínseca del objeto, y se definen como la magnitud aparente que mediríamos si el objeto se encontrara a una distancia de 10 parsecs de la Tierra (unos



Figura 2.3: Galaxia NGC1300, una espiral barrada, coherente con nuestro conocimiento actual de la morfología de la Vía Láctea

32,6 años luz). Así, por ejemplo, en la banda V el Sol tiene una magnitud aparente de $-26,74$, pero una magnitud absoluta de $4,83$.

2.3. Estructura a gran escala de la Vía Láctea

La Vía Láctea es una galaxia del tipo SBc (espiral barrada). Sus componentes estructurales a gran escala son, por lo tanto, los que pueden corresponder a una galaxia de este tipo: un disco (dividido en dos sistemas: disco fino y disco grueso), un halo esferoidal que envuelve el disco, un bulbo en la parte central del disco, una barra atravesando el bulbo, los brazos espirales, y posiblemente un anillo.

En la sección 3.4 daremos los detalles de las distribuciones de densidad estelar de cada uno de los componentes citados. Sin embargo, a continuación sigue una breve descripción de cada componente.

Disco

El disco es la componente más masiva de una galaxia espiral, y por lo tanto conocer bien su morfología y distribución de masa es fundamental. Inicialmente se creía que la densidad estelar del disco variaba con el cuadrado de una secante hiperbólica a lo largo del eje z , puesto que es el resultado teórico para un disco isoterma autogravitante, pero actualmente se sabe que

se ajusta mejor a una doble exponencial: una exponencial en función del radio galactocéntrico, multiplicada por otra exponencial en función de la altura sobre el plano. Su longitud de escala oscila, según los autores, entre 1 y 5 kiloparsecs, y su escala de altura está entre 125 y 475 parsecs (según la población estelar y la longitud de onda observada). El Sol se encuentra inmerso en el disco, a unos 8 kiloparsecs del centro y 15 parsecs sobre el plano (Hammersley et al. 1995).

En los años 80 se postuló la hipótesis de que el disco estaba realmente formado por dos sistemas diferentes: el disco fino que ya se conocía, y un disco grueso adicional (Gilmore and Reid 1983) para explicar las cuentas estelares en el polo Sur galáctico. Este disco estaría formado por estrellas menos metálicas y sus escalas de altura y longitud son mayores, aunque su densidad estelar en el plano es unas 10 veces menor que la del disco fino. Su función de luminosidad es más cercana a la del halo que a la del disco fino.

Hoy en día conocemos que el disco galáctico (tanto el fino como el grueso) no es un disco perfecto, sino que tiene ciertas irregularidades. Las irregularidades que se tienen en cuenta en el modelo que presentamos son:

- El alabeo (*“warp”*), que causa que el disco se curve de forma no simétrica en altura. Un modelo exitoso para describirlo es el de los “anillos inclinados”, según el cual el disco se puede dividir en varios anillos concéntricos, cada uno de los cuales tiene una inclinación respecto del plano que varía según su radio galactocéntrico, y formándose una línea de nodos común (ver la figura 2.4)
- El ensanchamiento (*“flare”*), que causa que la altura de escala se vaya incrementando con el radio galactocéntrico
- El déficit central, que conduce a una densidad estelar y del gas en la zona central del disco menor a la que cabría esperar según la ley doble exponencial.

Bulbo

En una galaxia espiral, el bulbo es el exceso de luz y estrellas con respecto a la extrapolación de un disco exponencial hacia las zonas centrales del mismo. Este componente presenta algunas similitudes con las galaxias elípticas: carencia de gas y de formación estelar, población estelar vieja, alta metalicidad, etc.

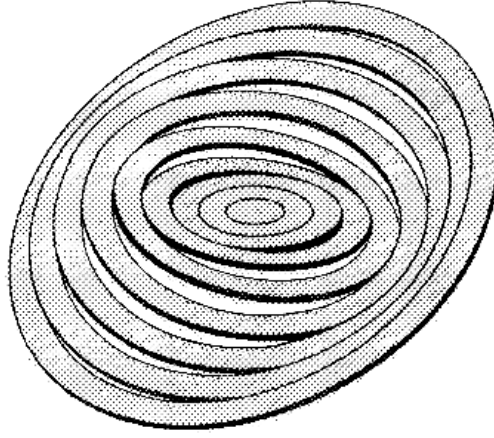


Figura 2.4: Modelo de anillos inclinados como descripción del alabeo galáctico (Rogstad et al. 1974)

Una de las primeras expresiones que se usaron para su distribución de densidad estelar fue propuesta por Bahcall (1986). En este primer modelo, el bulbo es a base de elipsoides concéntricos axisimétricos, con el eje menor perpendicular al plano. Actualmente, sin embargo, hay evidencias de que el bulbo no es, realmente, una estructura axisimétrica, sino que presenta triaxialidad. Además, su semieje mayor no está orientado en la línea de visión Sol-centro galáctico, sino que aparece con un ángulo respecto de nosotros, lo cual redundaría en una asimetría observable en las cuentas estelares para longitudes positivas respecto de longitudes negativas.

Barra larga

En la segunda mitad del siglo XX se empezó a sugerir la posibilidad que la Vía Láctea fuera una galaxia barrada. En los años 90 se empezaron a encontrar evidencias observacionales que apoyaban esta hipótesis cada vez con más fuerza. Algunos de estos avances fueron propuestos, justamente, desde el grupo de morfología y dinámica de la Vía Láctea del IAC, como por ejemplo en Hammersley et al. (1994), en el cual se usaban los datos del TMGS para analizar los picos del plano en $l = 27^\circ$ y $l = 21^\circ$, que se explicarían por la presencia de una barra (en concreto, el de $l = 27^\circ$ como una zona de formación estelar coincidiendo con el final de la barra). La existencia de la

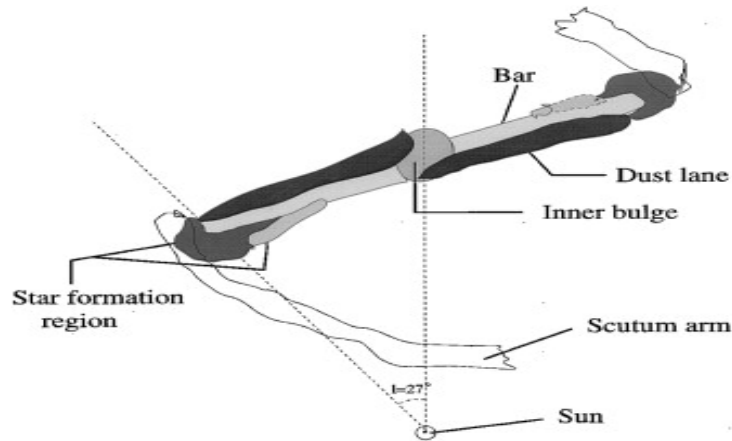


Figura 2.5: Esquema de la barra, con una banda de polvo que explicaría la mayor extinción en longitudes negativas, y regiones de formación estelar en ambos extremos de la barra (López-Corredoira et al. 1999)

barra se confirmó en la década de los 2000, al aparecer cartografiados más sensibles, como GLIMPSE (ver, por ejemplo, Benjamin et al. (2005)).

Muchos autores han utilizado el nombre “barra” para referirse al bulbo triaxial, y aún hoy en día hay cierta confusión en la terminología. En este trabajo utilizamos la nomenclatura “barra” para designar la barra larga, estructura diferenciada del bulbo triaxial.

En el rango de longitudes galácticas donde está presente la barra se puede observar una asimetría entre las cuentas estelares correspondientes a longitudes positivas y a longitudes negativas. Esto se explicaría debido al efecto de la geometría de la barra principalmente, sumado al efecto de una banda de polvo que precede a la barra para longitudes negativas (ver figura 2.5).

La barra de la Vía Láctea, según nuestros conocimientos actuales, tiene una morfología alargada y plana, con unas dimensiones aproximadas de $7,8 \text{ kpc} \times 1,2 \text{ kpc} \times 0,2 \text{ kpc}$, y estaría orientada respecto al Sol con un ángulo de posición de 43° (López-Corredoira et al. 2007).

Brazos espirales

Los brazos espirales son, quizá, uno de los elementos más espectaculares en cualquier imagen de una galaxia espiral. El origen de la estructura espiral aún no está bien determinada, aunque hay dos hipótesis predominantes. La

primera es que los brazos espirales surgen por una onda de densidad (Lin and Shu 1966), que rota más lentamente que las estrellas y el gas (y por lo tanto las estrellas que hay en los brazos no serían siempre las mismas, sino que entran y salen de ellos). La segunda teoría es la de la formación estelar estocástica auto-propagada (SSPSF), según la cual los brazos aparecen como efecto secundario de una actividad de formación estelar que se va propagando a zonas contiguas de la galaxia, unido al efecto de la rotación diferencial (Mueller and Arnett 1976).

En el caso de la Vía Láctea, debido a nuestra posición en el plano se convierten en una característica difícil de observar. Si bien es cierto que aparecen claramente en los mapas del gas obtenidos mediante técnicas radio-astronómicas, lo cierto es que a partir de estas medidas es difícil determinar la morfología exacta de los mismos. Aún así, se han realizado varios intentos usando diferentes fuentes (distribución de regiones HII, distribución del gas neutro, cuentas estelares, distribución de los electrones libres en la galaxia, etc). Una de los temas más controvertidos acerca de este componente estructural es acerca de si la Vía Láctea tiene dos o cuatro brazos.

Halo

El halo es una componente esferoidal, que se puede observar tanto en galaxias espirales como elípticas, y sigue una ley $r^{1/4}$ de De Vaucouleurs. Es una componente con una baja densidad estelar, y que supone un pequeño porcentaje de la masa total de la galaxia, materia oscura aparte.

El halo está poblado por estrellas viejas y metalicidades bajas, así como por cúmulos globulares (en el caso de la Vía Láctea, entre 150 y 160 cúmulos globulares). A diferencia del disco, las órbitas son muy elongadas, y está libre de polvo.

Capítulo 3

Modelo de cuentas estelares de la Vía Láctea

Como se ha comentado, el objetivo principal del presente Trabajo de Fin de Máster es construir un modelo de cuentas estelares de la Vía Láctea, mediante el cual se puedan obtener predicciones del número de estrellas que se observarían en una dirección determinada del cielo si la galaxia tuviera una determinada estructura. Posteriormente estas predicciones se pueden comparar con los datos observacionales disponibles a fin de ajustar los parámetros del modelo, así como confirmar o descartar hipótesis acerca de la morfología de nuestra galaxia.

La predicción del número de estrellas en un ángulo sólido Ω con magnitud aparente inferior a m en las coordenadas galácticas (l, b) que es consistente con un determinado modelo de la galaxia la obtenemos mediante la ecuación fundamental de la estadística estelar (von Seeliger 1898):

$$A(m, l, b) = \Omega \int_0^\infty r^2 \rho(r, l, b) \Phi(M) dr \quad (3.1)$$

donde r es la distancia heliocéntrica, ρ es la densidad espacial de estrellas en función de la posición en la galaxia, y $M(m, r)$ es la magnitud absoluta que corresponde a una estrella que se observa con magnitud aparente m y que se encuentra a r parsecs de distancia,

$$M(m, \vec{r}) = m - 5 \log_{10}(r) - A_\nu(\vec{r}) + 5 \quad (3.2)$$

donde $A_\nu(\vec{r})$ es el valor de la extinción en la línea de visión desde el Sol hasta el punto $\vec{r} = (r, l, b)$, y que se describe en la sección 3.2. Finalmente, Φ es la

integral hasta magnitud M de la función de luminosidad normalizada $\phi(M)$

$$\Phi(M) = \int_{-\infty}^M \phi(M') dM' \quad (3.3)$$

El modelo que hemos implementado es multibanda, y con la configuración por defecto soporta el cálculo de las cuentas estelares en los filtros B, V, J, H, K, 12 micras y 25 micras. Es posible cambiar las funciones de luminosidad que usa el modelo, con lo cual se puede extender a otras bandas.

3.1. Conversión a coordenadas galactocéntricas

Los datos observacionales de cuentas estelares suelen estar disponibles en coordenadas galácticas, que es un sistema esférico bidimensional centrado en el Sol. Sin embargo, la definición de la morfología de los componentes de la Vía Láctea necesitamos hacerla en un sistema tridimensional, y por cuestiones de simetría es preferible que su origen esté en el centro galáctico.

Para pasar de coordenadas galácticas (l, b) más la distancia heliocéntrica r a coordenadas cartesianas galactocéntricas usamos las siguientes relaciones

$$\begin{aligned} x &= R_{\odot} - r \cos b \cos l \\ y &= r \cos b \sin l \\ z &= r \sin b \end{aligned} \quad (3.4)$$

donde R_{\odot} es la distancia del Sol al centro de la galaxia, y es uno de los parámetros configurables del modelo. Se ha elegido $R_{\odot} = 8,0$ kpc como valor por defecto.

Además, en algunos casos la simetría hace más conveniente usar coordenadas cilíndricas galactocéntricas, que obtenemos con

$$\begin{aligned} R_c &= \sqrt{x^2 + y^2} \\ \theta_c &= \arctan \frac{y}{x} \\ z_c &= z \end{aligned} \quad (3.5)$$

3.2. Extinción

La inclusión de los efectos de la extinción interestelar en el cálculo de la magnitud absoluta a partir de la magnitud aparente es esencial si se quiere obtener un modelo realista, siendo su efecto menos significativo en longitudes de onda mayores (infrarrojo, radio) que en longitudes de onda más energéticas (visible, etc.).

La fracción de gas y, principalmente, de polvo causa una absorción en la luz estelar a medida que ésta atraviesa el medio interestelar. La intensidad específica I después de atravesar el medio es

$$I_\nu(s) = I_\nu(0)e^{-\tau_\nu} \quad (3.6)$$

donde τ_ν es la profundidad óptica,

$$\tau_\nu = \int_0^s \alpha_\nu(s')ds' = \int_0^s \kappa_\nu(s')\rho(s')ds' \quad (3.7)$$

Entonces, el aumento A_ν en magnitudes provocado por una profundidad óptica τ_ν , que aparecía en la ecuación (3.2), es

$$A_\nu = -2,5 \log_{10} e^{-\tau_\nu} = 1,086\tau_\nu \quad (3.8)$$

y por lo tanto éste es el término que debemos añadir a la relación para convertir magnitudes aparentes en magnitudes absolutas.

En numerosos modelos galácticos, como el modelo SKY (ver Wainscoat et al. (1992)), o el del observatorio de Besançon (en Robin et al. (2003)), se asume que el coeficiente de absorción κ_ν es constante, y que por lo tanto la extinción está determinada por el valor de la columna de materia difusa (gas y polvo) a lo largo de la línea de visión,

$$A_\nu = 1,086\kappa_\nu \int_0^s \rho_d(s')ds' = a_\nu(0) \int_0^s \frac{\rho_d(s')}{\rho_d(0)}ds' \quad (3.9)$$

El modelo de Besançon usa un valor normalización de $a_\nu(0) = 0,7 \text{ mag kpc}^{-1}$, y el modelo SKY usa un valor de $a_\nu(0) = 0,62 \text{ mag kpc}^{-1}$. Ambos trabajos usan una aproximación muy simple de la extinción, que se modela únicamente como un disco de materia difusa con una distribución exponencial de densidad.

En el presente Trabajo de Fin de Máster hemos decidido usar un modelo más complejo para la extinción, a fin de obtener resultados más realistas. Se

ha optado por el modelo de extinción de Drimmel et al. (2003), que usa como restricciones las observaciones de la emisión difusa en el infrarojo cercano procedente del polvo interestelar. Se usa como base del mapa de extinción el modelo de distribución del polvo de Drimmel y Spergel (2001), que contiene tres elementos estructurales principales: un disco con alabeo (“*warp*”) y con un gradiente radial de temperatura, los brazos espirales y un segmento del brazo local.

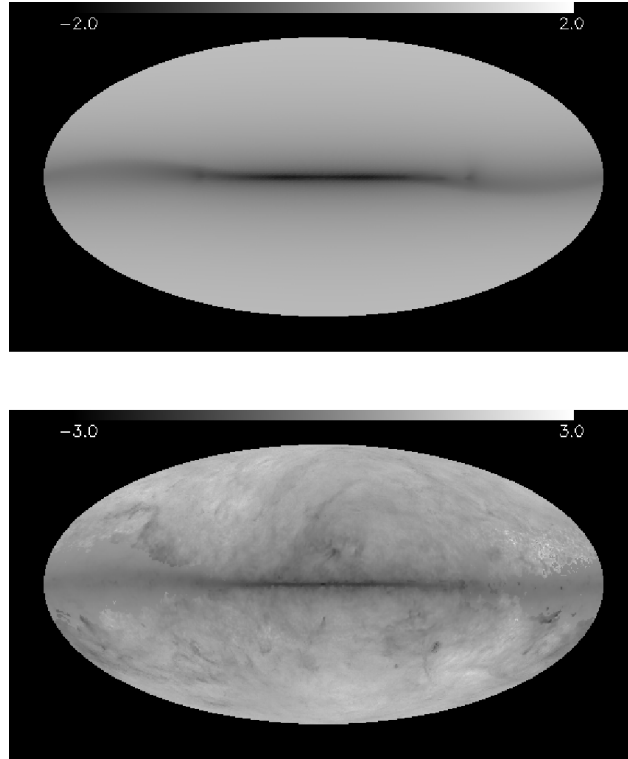


Figura 3.1: Mapa de cielo con la extinción galáctica total en el modelo usado Drimmel et al. (2003). La subfigura superior muestra la predicción de la extinción según el modelo de tres componentes, y la subfigura inferior contiene, además, el reescalado según los datos de DIRBE

Además, el modelo de Drimmel incluye un mecanismo para refinar la distribución de polvo a escalas angulares pequeñas, usando las observaciones en FIR. Se usan los residuales entre el modelo y los datos de $240\mu\text{m}$ de DIRBE para corregir el modelo. La corrección aplicada consiste en un reescalado de

la densidad del polvo a lo largo de toda la línea de visión. Este reescalado se aplica en el componente que necesita de un reescalado menor para cumplir la restricción observacional. En la figura 3.1 se puede observar el mapa en todo el cielo de la extinción total teniendo en cuenta sólo el modelo tridimensional, y teniendo en cuenta el efecto del reescalado.

El autor proporciona los datos del modelo de extinción, así como rutinas en Fortran para su acceso. Como en el presente Trabajo de Fin de Máster se decidió realizar la implementación del modelo de cuentas estelares usando el lenguaje Python, se ha procedido a reimplementar en este lenguaje las rutinas para el cálculo de la extinción usando los datos de Drimmel. En concreto, el procedimiento es el siguiente:

1. A partir de las coordenadas tridimensionales galactocéntricas, calcular el vóxel que contiene los datos acerca de la extinción en la celda de interés
2. Se aplica una interpolación trilineal para obtener el valor de la extinción en las coordenadas requeridas
3. A partir de las coordenadas galácticas (l, b) se calcula el mapa en las coordenadas de DIRBE
4. Se obtiene el factor de reescalado correspondiente al píxel de DIRBE obtenido, y se reescala el valor de la extinción de la componente apropiada

Finalmente, se ha implementado la conversión entre la extinción en la banda V al resto de bandas, usando los coeficientes de Rieke and Lebofsky (1985). Por lo tanto, en el modelo de cuentas estelares que se ha creado en el presente Trabajo de Fin de Máster disponemos de los datos tridimensionales de la extinción en las bandas U, B, V, R, I, J, H, K, L, M y N, así como en las de 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5, 12, 12.5 y 13 micras.

3.3. Función de luminosidad

La función de luminosidad $\phi(M)$ es la función de distribución de probabilidad de la magnitud absoluta de las estrellas pertenecientes a un grupo dado. Para nuestro objetivo de construir un modelo de cuentas estelares esta información es crucial, puesto que queremos obtener sólo el número de

Artículo	Banda	Origen	Descripción
(Wielen 1974)	v	Gliese	Función de luminosidad de la vecindad solar
(Eaton et al. 1984)	K	SAAO y TCS	Función de luminosidad del disco
(Wainscoat et al. 1992)	B, V, J, H, K, 12 μ m y 25 μ m	Mod. SKY	Funciones de luminosidad del disco, bulbo, brazos, halo y anillo molecular sintéticas
(López-Corredoira et al. 1997)	K	TMGS	Función de luminosidad del bulbo
(López-Corredoira et al. 2005)	K	2MASS	Función de luminosidad del bulbo
(Holtzman et al. 1989)	V, I	HST	Función de luminosidad de las estrellas de baja masa del bulbo
(Buser et al. 1999)	v	Basel	Función de luminosidad de disco fino, disco grueso y halo
(Mezger et al. 1999)	K	ESO-MPG	Función de luminosidad de los 30 pc centrales de la galaxia
(Reylé and Robin 2001)	v	DENIS	Función de luminosidad del disco fino y disco grueso
(Reid et al. 2002)	v	PMSU	Función de luminosidad de enanas M en la vecindad solar
(Rojo and Ruiz 2003)	v	Calán-ESO catalog	Función de luminosidad de enanas de la vecindad solar
(Zheng et al. 2004)	i	HST	Función de luminosidad de enanas M del disco
(Bochansky et al. 2010)	r, J	SDSS	Función de luminosidad y función inicial de masa de estrellas enanas del disco

Tabla 3.1: Literatura sobre funciones de luminosidad de la Vía Láctea

estrellas que se observarían hasta cierta magnitud aparente, a fin de poder comparar con datos observacionales (limitados en magnitud). Por lo tanto, en la integral a lo largo de la línea de visión sólo debemos tener en cuenta aquellas estrellas cuya magnitud absoluta se transforme en la magnitud aparente deseada una vez tenidos en cuenta los efectos de la distancia y de la extinción.

En el contexto del presente Trabajo de Fin de Máster, se ha hecho una búsqueda en la literatura de funciones de luminosidad recientes de los varios componentes de la Vía Láctea y en diferentes bandas (ver tabla 3.1). Sin embargo, la mayor parte de los artículos más recientes sólo tienen en cuenta la fracción de estrellas de baja masa ($0,1 M_{\odot} < M < 0,8 M_{\odot}$), y por lo tanto no son adecuadas a nuestros propósitos.

A fin de poder tener en cuenta la variedad de funciones de luminosidades disponibles, y de poder analizar el comportamiento de cada una respecto a los datos observacionales, el modelo Galiac se ha implementado modularmente, de forma que es sencillo cambiar la función de luminosidad que usa. Queda pendiente mejorar esta parte, y conseguir un conjunto de funciones de luminosidad en todos los filtros más actualizadas.

Las funciones de luminosidad por defecto que se han elegido son las de Wainscoat et al. (1992), por ser multibanda, y en el caso concreto de la banda K se usa la función de luminosidad de Eaton et al. (1984) para el disco y de López-Corredoira et al. (2005) para el bulbo.

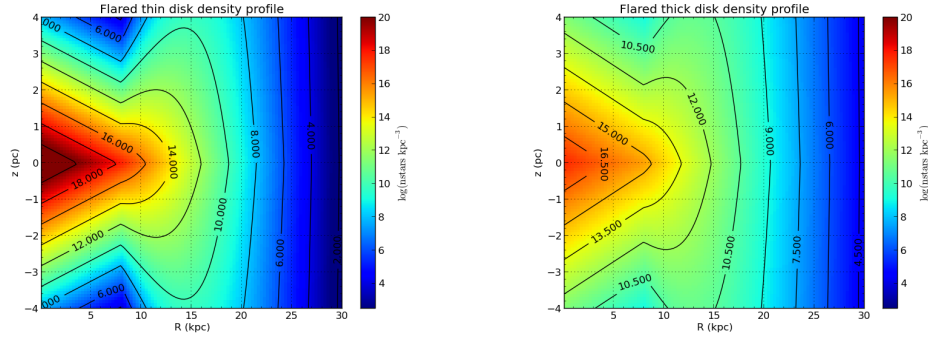
3.4. Funciones de densidad

3.4.1. Disco

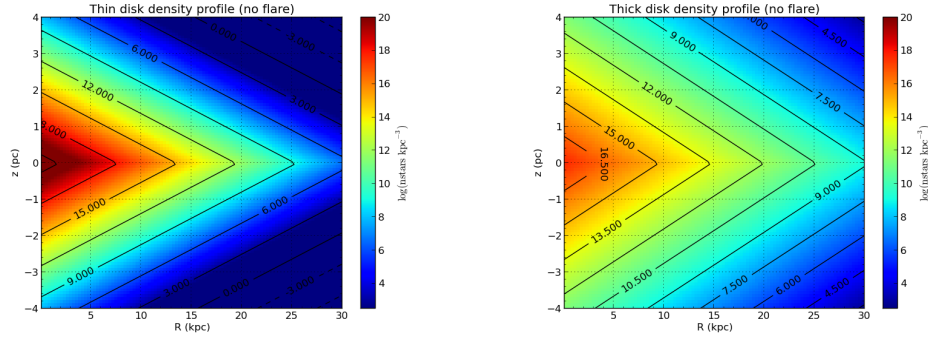
La base para el modelo del disco implementado en Galiac (tanto para el disco fino como para el disco grueso) es la presentada en López-Corredoira et al. (2002): una distribución de densidad en forma de doble exponencial con un ensanchamiento (“*flare*”) en la escala de altura

$$\rho(R, z) = \rho_{\odot} \exp\left(-\frac{R - R_{\odot}}{h_R}\right) \exp\left(-\frac{|z|}{h_z(R)}\right) \left(\frac{h_z(R_{\odot})}{h_z(R)}\right) \quad (3.10)$$

donde R_{\odot} es la distancia del Sol al centro galáctico, h_R es la escala de longitud y $h_z(R)$ es la escala de altura. A causa del ensanchamiento, esta escala de altura no es constante, sino que varía exponencialmente según el radio



(a) Distribución de densidad del disco fino, con ensanchamiento (b) Distribución de densidad del disco grueso, con ensanchamiento



(c) Distribución de densidad del disco fino, sin ensanchamiento (d) Distribución de densidad del disco grueso, sin ensanchamiento

Figura 3.2: Distribución de densidad del disco

galactocéntrico:

$$h_z(R) = h_z(R_\odot) \exp\left(\frac{R - R_\odot}{h_{R,flare}}\right) \quad (3.11)$$

En la figura 3.2 se puede observar la función de densidad del disco fino y grueso, así como comparar el efecto del ensanchamiento descrito.

Posteriormente se modificó el modelo para añadir también el efecto del alabeo y del déficit de estrellas central.

Alabeo

Para el alabeo se usó un modelo de anillos concéntricos inclinados entre sí, cuya inclinación varía con la distancia galactocéntrica. En Galiac se ha experimentado con dos formulaciones diferentes del alabeo, la presentada en López-Corredoira et al. (2002) y la de Levine et al. (2006a) (que corresponde al alabeo observado en la distribución del gas). En el primer caso se tiene en cuenta un sólo modo de frecuencia, el $m = 1$, y la amplitud varía con el radio galactocéntrico según una ley de potencias,

$$z_w(R, \theta) = C_w R^{\epsilon_w} \sin(\theta - \theta_w) + h_\odot \quad (3.12)$$

donde z_w es la elevación del disco sobre el plano paralelo a $b = 0$, C_w es una constante de proporcionalidad, R es el radio galactocéntrico en parsecs, ϵ_w es el exponente de la ley de potencias, θ_w es el ángulo acimutal que define la línea de nodos, y h_\odot es la altura del Sol sobre el plano. Esta expresión sólo es válida para $R < 15$ kpc, puesto que cuando se extrapola para radios galactocéntricos mayores el alabeo que produce crece excesivamente.

En cambio, en la segunda implementación del alabeo se usan tres modos de Fourier,

$$z_w(R, \theta) = W_0(R) + W_1(R) \sin(\theta - \theta_1) + W_2(R) \sin(2\theta - \theta_2) + h_\odot \quad (3.13)$$

donde θ_i son las fases de cada modo y W_i las amplitudes, definidas por

$$W_i(R) = k_0 + k_1(R - R_k) + k_2(R - R_k)^2 \quad (3.14)$$

El primer alabeo es en forma de S, que crece muy rápidamente con el radio galactocéntrico debido a su comportamiento como ley de potencias. En cambio, el segundo, al incluir tres modos, tiene una forma más compleja y crece más lentamente puesto que la amplitud de cada modo varía con el cuadrado del radio galactocéntrico. En la figura 3.3 se puede comparar el comportamiento de ambos alabeos.

Déficit central de estrellas

Aproximadamente la mitad de las galaxias espirales barradas presentan un déficit de estrellas y gas (o “agujeros”) en la parte central del disco. En la Vía Láctea se observa un déficit central en la componente gaseosa del disco y también en la componente estelar, aunque de forma más controvertida.

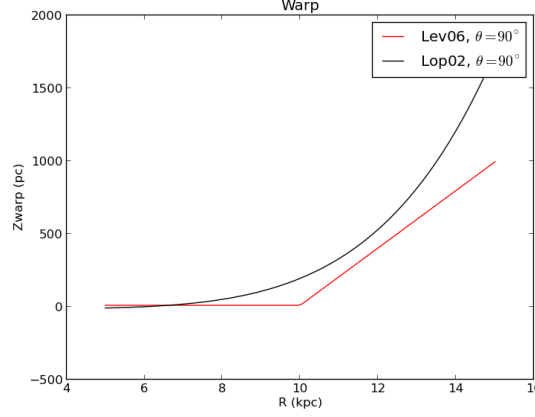


Figura 3.3: Variación de la altura del disco en función del radio galactocéntrico para un ángulo acimutal de $\theta = 90^\circ$, según las dos modalidades de alabeo usadas en Galiac

En el modelo Galiac se ha tenido en cuenta el agujero en la distribución de densidad estelar del disco fino y del disco grueso. La expresión analítica que usamos es la de López-Corredoira et al. (2004), que combina una modificación en la exponencial de la ecuación (3.10) con una expresión polinómica para la escala de altura en la región interna,

$$\rho_{hole}(R, z) = \rho(R, z) \exp \left(\frac{H_{R,hole}}{R_\odot} - \frac{H_{R,hole}}{R} \right) \quad (3.15)$$

$$h_z(R < R_{hole}) = C_{h,0} + C_{h,1}(R - R_{hole}) + C_{h,2}(R - R_{hole})^2 \quad (3.16)$$

Sumario del disco

Teniendo en cuenta todos los términos que hemos presentado en las anteriores subsecciones, el modelo final del disco que se ha implementado en Galiac corresponde a la formulación siguiente (usamos la notación f, g para los parámetros y expresiones que son equivalentes entre el disco fino y el disco grueso, pero cuyo valor numérico cambia según si se trata del primero o del

segundo):

$$\rho_{d\{f,g\}}(R, z) = \rho_{\odot} e^{\left(-\frac{R-R_{\odot}}{h_{R\{f,g\}}}\right)} e^{\left(\frac{H_{R,hole}}{R_{\odot}} - \frac{H_{R,hole}}{R}\right)} e^{\left(-\frac{|z-z_w|}{h_{z\{f,g\}}(R)}\right)} \left(\frac{h_{z\{f,g\}}(R_{\odot})}{h_{z\{f,g\}}(R)}\right) \quad (3.17)$$

$$h_{z\{f,g\}}(R) = \begin{cases} h_{z\{f,g\}}(R_{\odot}) \sum_{i=0}^2 C_{h,i} (R - R_{hole})^i, & \text{si } R < R_{hole} \\ h_{z\{f,g\}}(R_{\odot}) & \text{si } R_{hole} < R < R_{flare} \\ h_{z\{f,g\}}(R_{\odot}) \exp\left(\frac{R-R_{\odot}}{h_{R,flare}}\right), & \text{si } R > R_{flare} \end{cases} \quad (3.18)$$

$$z_{w\{lop\}} = C_w R^{\epsilon_w} \sin(\theta - \theta_w) + h_{\odot} \quad (3.19)$$

$$z_{w\{lev\}} = \sum_{m=0}^2 \left(\sum_{i=0}^2 k_{m,i} (R - R_m)^i \right) \sin(m\theta - \theta_m) \quad (3.20)$$

Los parámetros que se usan en el modelo son configurables por el usuario, y también se pueden ajustar a datos observacionales. Sin embargo, el modelo incorpora unos valores por defecto de estos parámetros, listados en la tabla 3.2.

En la figura 3.4 se pueden ver las distribuciones de densidad del disco fino y del disco grueso con todas las modificaciones añadidas (ensanchamiento, alabeo y déficit central).

3.4.2. Bulbo

Para la función de distribución de densidad del bulbo se han usado los resultados de López-Corredoira et al. (2005), donde se usó un método de inversión de la ecuación integral fundamental de la estadística estelar con datos de 2MASS.

La expresión usada para la función de densidad corresponde a un bulbo triaxial con forma de caja (“*boxy*”), en lugar de ser elipsoidal. El sistema de referencia que usamos es el asociado a los semiejes de la función de distribución de densidad, que al no ser axisimétrica está rotada respecto al sistema de coordenadas cartesiano galactocéntrico. En este caso sólo contemplamos

Parámetro	Valor
$\rho_{\odot\{f\}}$	$55,0 \times 10^6 \star \cdot \text{kpc}^{-3}$
$\rho_{\odot\{g\}}$	$4,95 \times 10^6 \star \cdot \text{kpc}^{-3}$
R_{\odot}	8,0 kpc
h_{\odot}	15 pc
$H_{R\{f\}}$	1,97 kpc
$H_{R\{g\}}$	3,5 kpc
$H_{R,hole}$	3,74 kpc
$h_{z\{f\}}(R_{\odot})$	0,285 kpc
$h_{z\{g\}}(R_{\odot})$	0,631 kpc
R_{hole}	8,0 kpc
$C_{h,0}$	1,0
$C_{h,1}$	$0,21 \text{ kpc}^{-1}$
$C_{h,2}$	$0,056 \text{ kpc}^{-2}$
$H_{R,flare}$	4,5 kpc
C_w	$2,1 \times 10^{-19} \text{ pc}$
ϵ_w	5,25
θ_w	$-5,0^\circ$

Tabla 3.2: Valores por defecto de los parámetros del disco

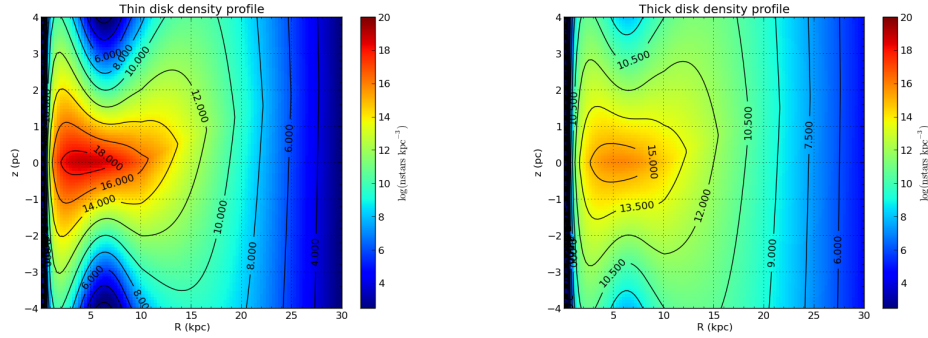
una rotación en el eje z , con lo cual dicha rotación queda completamente definida con un sólo ángulo α_{bulge} . Para la función de distribución de densidad usamos la expresión paramétrica

$$\rho_{bulge}(t) = \rho_{bulge}(0) \exp\left(-\frac{t}{h_b}\right) \quad (3.21)$$

donde $\rho_{bulge}(0)$ es la densidad central, h_b es el parámetro de escala, y la variable paramétrica t sigue la expresión

$$t = \left[x_b^n + \left(\frac{y_b}{r_y}\right)^n + \left(\frac{z_b}{r_z}\right)^n \right]^{1/n} \quad (3.22)$$

en la cual la relación entre los semiejes es $[1 : r_y : r_z]$, y (x_b, y_b, z_b) son las coordenadas (x, y, z) rotadas al sistema de coordenadas del bulbo. Para $n = 2$ los isocontornos de densidad corresponden a elipsoides. Para $n = 4$, que es el valor que hemos usado, corresponden a la forma de caja.



(a) Distribución de densidad total del disco fino (b) Distribución de densidad total del disco grueso

Figura 3.4: Distribución de densidad total del disco

Parámetro	Valor
α_{bulge}	27°
$\rho_{bulge}(0)$	$6,6 \times 10^9 \star \cdot \text{kpc}^{-3}$
h_b	0,640 kpc
r_y	0,49
r_y	0,40

Tabla 3.3: Valores por defecto de los parámetros del bulbo

Parámetro	Valor
α_{bar}	$43^\circ \pm 7^\circ$
$\rho_{bar}(0)$	$3,0 \times 10^9 \star \cdot \text{kpc}^{-3}$
len_{bar}	3,9 kpc
$\sigma_{y,bar}$	0,39 kpc
$\sigma_{z,bar}$	0,117 kpc

Tabla 3.4: Valores por defecto de los parámetros de la barra

En la tabla 3.3 se encuentran los parámetros por defecto usados en el modelo Galiac para la función de distribución de densidad del bulbo.

3.4.3. Barra

Aproximadamente la mitad de las galaxias espirales tienen una barra central que atraviesa el bulbo, y en cuyos extremos empiezan los brazos espirales. A causa de los efectos de la alta extinción en el plano y de nuestra posición en la galaxia es difícil observar esta componente, y durante unos años hubo controversia acerca de la naturaleza de algunos de los picos de cuentas estelares en la zona de $-30^\circ \leq l \leq 30^\circ$ en el plano, ya que podrían estar causados por una barra, los brazos espirales o un anillo estelar. Finalmente se corroboró la hipótesis de la barra galáctica, y por lo tanto ha sido incluida

en el modelo galáctico que presentamos.

En López-Corredoira et al. (2007) se usaron los datos de varias fuentes para trazar la barra. El resultado es una barra larga, contenida en el plano, abarcando un rango amplio de longitudes ($-14^\circ \leq l \leq +30^\circ$) pero mucho más compacto en latitud ($|b| < 1,5^\circ$). Esta componente está diferenciada del bulbo triaxial (que se extiende hacia latitudes $|b| < 10^\circ$) y presenta un ángulo de posición diferente (aunque no muy alejado), de $\alpha_{bar} = 43^\circ \pm 7^\circ$.

La expresión analítica que usamos para modelar la barra representa una distribución de densidad que es constante a lo largo del semieje mayor, y que está distribuido de forma gaussiana a lo largo de los semiejes menores:

$$\rho_{bar} = \rho_{bar}(0) \exp\left(\frac{y_{bar}^2}{2\sigma_{y,bar}^2}\right) \exp\left(\frac{z_{bar}^2}{2\sigma_{z,bar}^2}\right) \quad (3.23)$$

donde $\rho_{bar}(0)$ es la densidad central de la barra, $(x_{bar}, y_{bar}, z_{bar})$ son las coordenadas (x, y, z) rotadas al sistema de coordenadas de la barra, len_{bar} es la semi-longitud de la barra, y $\sigma_{y,bar}$ y $\sigma_{z,bar}$ son las anchuras de los dos semiejes menores. La expresión (3.23) la truncamos para que sólo sea válida para $|x_{bar}| \leq \text{len}_{bar}$, $|y_{bar}| \leq \sigma_{y,bar}$ y $|z_{bar}| \leq \sigma_{z,bar}$. En la tabla 3.4 se listan los valores por defecto de los parámetros de la barra.

3.4.4. Brazos espirales

Al igual que la barra, los brazos espirales de nuestra galaxia son una componente difícil de observar a causa de la alta extinción causada por el polvo, y de la posición del Sol en el plano.

Un observable directo de los brazos espirales es el listado de los puntos tangentes entre las líneas de visión desde el Sol y los propios brazos. Efectivamente, puesto que los brazos y el Sol están en el mismo plano, es difícil trazar la forma espiral pero en cambio los puntos tangentes a las líneas de visión deben aparecer como sobredensidades. En Englmaier and Gerhard (1999) se hace una compilación de varios artículos de la literatura que abordan esta cuestión, proporcionando datos ligeramente diferentes pero consistentes entre sí, y en otros artículos posteriores se revisita este listado, como en Francis and Anderson (2012) (ver tabla 3.5).

Si bien hay un buen consenso acerca de la posición de los puntos de tangencia, por ser un observable directo, esto no es así cuando se trata de la morfología en sí de los brazos. En concreto, el número de brazos espirales

	Longitud		
Tangencia	Inicio	Pico	Final
Orion-Perseo	58	66	
Sagitario	39	43	46
Scutum	26	29	34
2.5 kpc	19	20.5	22.5
Inicio Orion	14.5	15.5	17
Barra	12	0	-8.5
Inicio Centauro	-11.5	-13	-14.5
2 kpc	-17	-18.5	-20.5
3 kpc	-22	-25	-28
Norma	-34	-37	-40
Scutum Crux	-50	-52	-55
Orion-Sagitario	-63	-70	

Tabla 3.5: Longitud galáctica de los puntos de tangencia de los brazos espirales con las líneas de visión desde el Sol (Francis and Anderson 2012)

es uno de los aspectos controvertidos. Hay autores que defienden la hipótesis que la Vía Láctea tiene 4 brazos espirales, como por ejemplo Georgelin and Georgelin (1976) (que usó datos sobre la distribución de regiones HII), Taylor and Cordes (1993) (usando la distribución de electrones libres), Levine et al. (2006b) (usando datos sobre la distribución de hidrógeno neutro en el disco) o Wainscoat et al. (1992). En cambio, otros autores defienden la hipótesis de la existencia de 2 brazos espirales, como Bash (1981) (con los mismos datos que Georgelin usó para obtener 4 brazos), o Francis and Anderson (2012) (con datos combinados sobre la distribución de HI y de 2MASS).

En el presente trabajo se ha elegido un modelo de dos brazos simétricos, siguiendo una espiral logarítmica:

$$r(\alpha) = r_0 e^{-b(\alpha - \alpha_0)} \quad (3.24)$$

donde α es el parámetro que corresponde al ángulo acimutal, r_0 es el radio galactocéntrico de la espiral cuando $\alpha = \alpha_0$, y b es la tangente del ángulo de ataque (“*pitch angle*”).

La función de distribución de densidad de los brazos sigue el mismo razonamiento que Wainscoat et al. (1992): en el centro del brazo se corresponde

con la función de densidad del disco fino, pero incrementada en un factor f_{arm} y con una escala de altura diferente, y se extiende luego horizontalmente con una densidad constante igual a la del centro del brazo hasta un tamaño σ_{arm} .

Hay que tener en cuenta que, en la expresión (3.24), α no está limitado al rango $[0, 2\pi]$ sino que varía en el rango $(0, \infty)$. Para un mismo ángulo acimutal θ , los radios en los cuales pasa la espiral son

$$r = r_0 e^{-b(\theta - \alpha_0 + n \cdot 2\pi)} \quad (3.25)$$

donde n es el número de vueltas que ha dado la espiral hasta el punto en cuestión.

En general, cualquier coordenada cilíndrica (R, θ) está situada entre dos vueltas de la espiral (a no ser que coincida exactamente con el centro). Para encontrar el número de vuelta que se aproxima más al punto en cuestión, calculamos

$$n = \frac{\frac{\log \frac{r}{r_0}}{b} - \theta}{2\pi} \quad (3.26)$$

Esto nos da un valor real para n , que coincide con el número de vuelta en el caso de que la coordenada dada sea efectivamente un punto de la espiral. En caso contrario, elegimos el valor entero de n redondeado hacia arriba o hacia abajo, $n_s = \lfloor n \rfloor$ o $n_s = \lceil n \rceil$, de forma que se minimice la distancia $d_s(R, \theta)$ del punto hasta la espiral. Posteriormente usamos esta distancia, y en caso que sea inferior a σ_{arm} el punto se considera dentro del brazo espiral:

$$\rho_{arm}(R, \theta, z) = \begin{cases} f_{arm} \rho_{d\{f\}}(r_0 e^{-b(\theta - \alpha_0 + n_s \cdot 2\pi)}, z), & \text{si } d_s(R, \theta) \leq \sigma_{arm} \\ 0, & \text{si } d_s(R, \theta) > \sigma_{arm} \end{cases} \quad (3.27)$$

En la tabla 3.6 se encuentran los valores por defecto de Galiac para los parámetros de los brazos espirales, y en la figura 3.5 se puede ver gráficamente la función de distribución de densidad de los brazos en el plano XY y en el plano XZ.

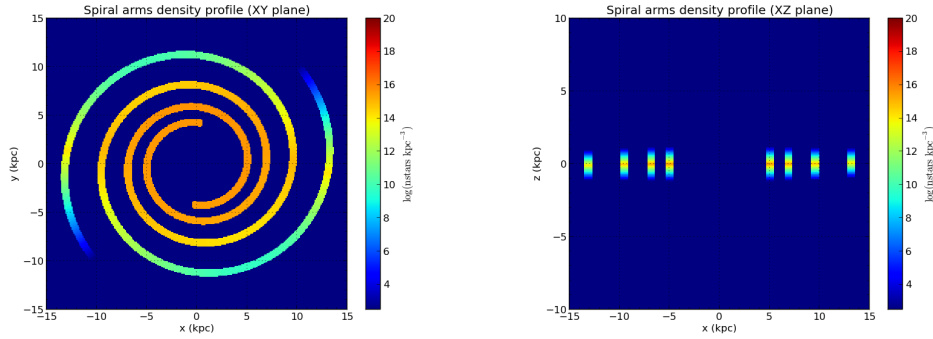
3.4.5. Halo

Finalmente, el último componente estructural de la Vía Láctea que incorpora el modelo Galiac es el halo. Éste se modela como un esferoide oblató, siguiendo una ley de De Vaucouleurs

$$\rho_{halo}(R, z) = \rho_{\odot} \frac{e^{(10.093(1 - (R_{sp}/R_{\odot})^{1/4}))}}{(R_{sp}/R_{\odot})^{7/8}} \quad (3.28)$$

Parámetro	Valor
f_{arm}	0,083
α_0	83,5°
r_0	4,2 kpc
Pitch angle	5,56°
σ_{arm}	0,212 kpc

Tabla 3.6: Valores por defecto de los parámetros de los brazos espirales



(a) Distribución de densidad de los brazos espirales en el plano XY (b) Distribución de densidad de los brazos espirales en el plano XZ

Figura 3.5: Distribución de densidad de los brazos espirales

donde R_{sp} es

$$R_{sp} = \sqrt{R^2 + 2,52z^2} \quad (3.29)$$

La expresión (3.28) se trunca para valores de $R < 1$ kpc para evitar la singularidad que se produce en el origen.

3.5. Ajustes de los parámetros estructurales

Uno de los objetivos del presente Trabajo de Fin de Máster es la realización de ajustes de los parámetros estructurales a datos observacionales procedentes de cartografiados del cielo.

Los ajustes se han realizado mediante el algoritmo de minimización de Levenverg-Marquardt (Marquardt 1963), que proporciona una convergencia

robusta y rápida hacia un mínimo local de χ^2 . Este algoritmo es una interpolación entre el algoritmo de Gauss-Newton (basado en un desarrollo en series de Taylor de la función objetivo) y el algoritmo de *steepest descent* (que usa los gradientes de los residuales para ir avanzando hacia el mínimo local), y generalmente está libre de los problemas asociados a estos dos algoritmos (en el caso del algoritmo de Gauss-Newton la falta de robustez, que lo conduce en ocasiones a diverger de la solución óptima, y en el caso del *steepest descent* su tendencia a zigzaguear en el entorno del mínimo, lo cual lo convierte en muy ineficiente en algunos problemas).

La estrategia que se ha seguido ha sido la de realizar varios ajustes por separado de los parámetros de componentes independientes, a fin de no sobrecargar la rutina de ajuste con demasiados parámetros libres. Por otra parte, los parámetros que están relacionados entre sí se han ajustado simultáneamente. Además, se han incorporado reglas de validez en cada ajuste para evitar que el algoritmo escoja valores de los parámetros que estén fuera de su rango válido.

Capítulo 4

Implementación

En esta sección se describe la implementación realizada del modelo descrito en el apartado anterior, y que junto al análisis de los datos ha constituido el grueso del presente Trabajo de Final de Carrera.

4.1. Gestión del proyecto

La construcción del modelo *Galiac* ha sido un proceso relativamente largo y complejo, y que además ha requerido un enfoque multidisciplinar. Ha sido necesaria la interacción con varios investigadores del Instituto de Astrofísica de Canarias para la captura de requerimientos, así como la consulta de numerosos artículos científicos, para luego plasmarlo en un código versátil y eficiente.

4.1.1. Planificación

En la figura 4.1 se puede ver la descomposición del proyecto en tareas y subtareas.

Al tratarse de un proyecto ejecutado enteramente por una sola persona no se procedió a realizar cronogramas detallados. Sin embargo, si se realizó una estimación del coste temporal de cada paquete de trabajo. Proyectando este coste a la dedicación que podía asumir el autor, el resultado fue la siguiente planificación:

- **Definición de proyecto:** 2 meses y medio

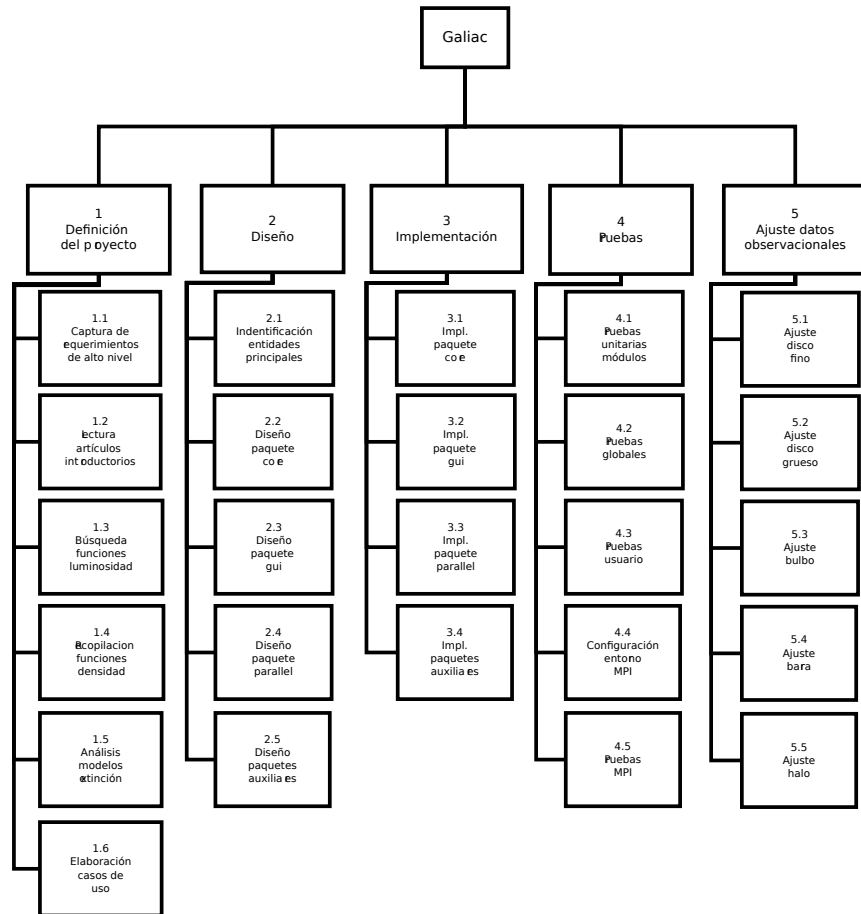


Figura 4.1: WBS (*Work Breakdown Structure*) del proyecto

- **Diseño:** 1 mes y medio
- **Implementación:** 2 meses y medio
- **Pruebas:** 3 semanas (dos de las cuales, para la configuración y pruebas en MPI)
- **Ajustes:** 1 mes

El desarrollo posterior del proyecto se mantuvo bastante fiel a esta previsión.

4.1.2. Gestión de la base de código y control de versiones

Un sistema de control de versiones es un repositorio de documentos que almacena también los cambios que se aplican a ellos. Hay varios sistemas de control de versiones disponibles, pero en general la mayor parte de ellos permiten trazar los cambios (qué cambió, quién y cuando lo cambió, y por qué motivo se introdujo el cambio), comparar versiones diferentes del documento, regresar a una versión antigua, o separar el desarrollo en varias ramas y poderlas mezclar posteriormente. Las herramientas de control de versiones son fundamentales en la gestión de la configuración del software y, por lo tanto, en la ingeniería del software.

Para gestionar la base de código del modelo *Galiac* hemos elegido *Git*. *Git* es un sistema de control de versiones distribuido, en el cual el código y el histórico de cambios no se almacenan en un único servidor centralizado sino que se replica cada vez que un usuario se lo descarga. Esta aproximación proporciona la ventaja que el repositorio es más robusto ante fallos de hardware, y que se facilita el trabajo simultáneo por parte de distintos grupos de desarrollo. En nuestro caso, hemos trabajado con una copia del repositorio en local, y además se ha publicado otra copia (sincronizada con la local) en el conocido servidor *GitHub*, a la cual se puede acceder en la *url* <https://github.com/jmolgo/galiac>.

4.2. Consideraciones de diseño

Para la implementación del modelo galáctico se ha intentado conseguir un diseño modular, flexible, reutilizable y fácilmente mantenible. Esto es

importante, puesto que uno de los propósitos del autor del presente TFC es la distribución del modelo para que sea utilizado en varias aplicaciones del ámbito académico, por varios equipos distintos y organizaciones ajenas.

4.2.1. Lenguaje de programación

El modelo Galiac se ha implementado en el lenguaje de programación Python. Python es un lenguaje interpretado, con un sistema de tipos de datos dinámico y gestión automática de la memoria. Se considera un lenguaje multiparadigma, puesto que soporta programación orientada a objetos, programación imperativa y programación funcional. Python proporciona, además, una librería estándar muy completa, que se complementa con un amplio abanico de librerías disponibles, muchas de ellas orientadas al cálculo científico.

La elección de Python como lenguaje de programación fue fuertemente condicionada por el cliente (usuario final), puesto que se deseaba incorporar el Modelo a otras aplicaciones escritas en el mismo lenguaje. Otro factor que influyó en el uso de Python fue que uno de los requerimientos del proyecto era dotar al modelo de varias interfaces (interfaz *web*, interfaz gráfica de escritorio, uso como librería, tarea de Iraf, etc.), y *Python* ofrece también las herramientas para ello.

El modelo Galiac usa extensivamente las librerías de Python **Numpy** y **Scipy**. **Numpy** es una librería que ofrece soporte para matrices y vectores, álgebra lineal, etc. Los vectores y matrices de **Numpy** permiten aplicar una operación matemática a la totalidad de sus componentes, con lo cual se evita la necesidad de realizar bucles explícitos para la implementación de los cálculos (lo cual es ineficiente en Python). Además, esta capacidad de operar de forma vectorial dota al lenguaje de una expresividad mucho mayor para los cálculos físicos, es decir, con menos líneas de código se consigue el mismo resultado que en otros lenguajes, y las sentencias se parecen mucho a las expresiones matemáticas que se quieren implementar.

Por otra parte, **Scipy** es una librería orientada a cálculos científicos, como por ejemplo estadística, integración numérica, ajuste de funciones por mínimos cuadrados, transformadas de Fourier, procesado de la señal, etc. En concreto, nosotros usaremos extensivamente los siguientes: paquetes de **Scipy**:

- **optimize**: Este paquete proporciona rutinas que implementan varias estrategias de optimización y de ajuste por mínimos cuadrados

- **interpolate:** Proporciona funciones para realizar interpolaciones de datos unidimensionales y bidimensionales mediante varios métodos
- **integrate:** Este subpaquete contiene la implementación de varios algoritmos de integración numérica (trapecios, Simpson, Romberg, etc)

4.2.2. Patrones GRASP

Los patrones de diseño **GRASP**, o “*General Responsibility Assignment Software Patterns*” (Larman 2005), son unas recomendaciones para asignar responsabilidades a las varias clases de un diseño orientado a objetos. En lugar de proporcionar diseños concretos que resuelven cierto problema, como es el caso de otros patrones de diseño, los patrones GRASP realmente sientan unas bases, más abstractas, que sirven de guía a la hora de repartir en clases concretas las funcionalidades y responsabilidades que se derivan de los requerimientos.

En la implementación del modelo Galiac hemos usado los siguientes patrones GRASP:

- **Creator:** Este patrón proporciona una guía para la decisión de cuál debe ser la clase responsable de crear los objetos de otra. En concreto, según el patrón *Creator* una clase B debería ser la responsable de crear una clase A si alguna de las siguientes condiciones es cierta:
 - Las instancias de B contienen o agregan por composición las instancias de A
 - Las instancias de B almacenan instancias de A
 - Las instancias de B realizan un uso intensivo de las instancias de A
 - Las instancias de B tienen la información necesaria para inicializar las instancias de A
- **High cohesion:** Según el criterio de alta cohesión, las clases deben estar lo más focalizadas posible entorno a un objetivo, y por lo tanto sus responsabilidades deben estar altamente relacionadas y agrupadas. Un sistema con alta cohesión es más legible y mantenible. En cambio, un sistema con baja cohesión, donde un elemento dado tiene demasiadas responsabilidades o donde una responsabilidad concreta está repartida por varios componentes, es difícil de comprender, reutilizar y mantener.

- **Low coupling:** Este criterio está directamente relacionado con el de alta cohesión. Un sistema con bajo acoplamiento es aquel que tiene pocas dependencias entre las clases, y en el cual un cambio en una clase determinada provoca un impacto bajo en el resto. Los sistemas que gozan de bajo acoplamiento son más fáciles de reutilizar y de mantener.
- **Information expert:** Es un criterio para determinar dónde delegar responsabilidades. La idea básica es que, dada una funcionalidad a implementar, se debe determinar cuál es la información necesaria para acometerla, y dónde está almacenada. La clase que debe implementar esta funcionalidad es la que contiene la mayor parte de la información requerida.
- **Polymorphism:** Usando polimorfismo, la responsabilidad de implementar las variaciones de un comportamiento debe recaer en los tipos en la cual esta variación ocurre.

4.2.3. Patrones GOF

“Patrones GOF (*Gang of Four patterns*)” es el nombre que habitualmente se le da al conjunto de patrones descritos en el libro *Design Patterns*, publicado por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (Gamma et al. 1994). Son un conjunto de una veintena de patrones, agrupados según el tipo: patrones creacionales, estructurales o de comportamiento. A diferencia de los patrones GRASP, éstos exponen soluciones óptimas a problemas más concretos.

En el presente TFC, hemos usado los siguientes:

- **Strategy:** es un patrón de comportamiento que permite elegir en tiempo de ejecución el comportamiento de un algoritmo.
- **Factory method:** es un patrón creacional que usa el concepto de las factorías para crear objetos sin especificar exactamente su clase.
- **Template method:** es un patrón de comportamiento que permite definir en una clase el esqueleto de un algoritmo, del cual algunos pasos se implementan en las clases derivadas.

4.3. Arquitectura de la aplicación

En la presente sección se detalla la arquitectura de la aplicación. Se ha utilizado el paradigma de **programación orientada a objetos**, identificando las entidades relevantes del dominio y encapsulándolas dentro de clases. Para la descripción de la arquitectura se usará el **model 4+1**.

4.3.1. Modelo 4+1

El **modelo 4+1** (Kruchten 1995) es un modelo de vista arquitectural desarrollada por Philippe Kruchten (el cual es también uno de los responsables del *Rational Unified Process*). Para describir la arquitectura de un sistema software este modelo usa múltiples vistas concurrentes a fin de expresar las características del sistema bajo enfoques diferentes. No hay una vista que sea preponderante respecto a las otras, sino que todas expresan información igualmente importante de aspectos diferentes, son complementarias y pueden solaparse en algunos aspectos (de ahí la expresión de “vistas concurrentes”). El objetivo de esta vista arquitectural es mostrar de forma separada los requerimientos funcionales de los no funcionales, así como las necesidades e intereses de los varios *stakeholders* del proceso: usuarios finales, desarrolladores, administradores de sistema, etc.

Tal como se expresa en la figura 4.2, las vistas propuestas por el modelo 4+1 son la *vista lógica*, la *vista de desarrollo*, la *vista de proceso* y la *vista física*, complementadas por los *escenarios*, o casos de uso, de los cuales emanan el resto de vistas:

- **Vista lógica:** Esta vista describe el modelo de objetos del diseño. Es la que debe soportar los requerimientos funcionales. Usando los principios de abstracción, encapsulación y herencia se descompone el sistema en un conjunto de *clases y objetos*, extraídos del dominio del problema. Para esta vista se suelen usar los diagramas UML de clases.
- **Vista de proceso:** Esta vista aporta los aspectos dinámicos del sistema. Se concentra en el comportamiento del sistema en tiempo de ejecución, explicando los varios procesos y su comunicación, así como los aspectos relacionados con la concurrencia y el rendimiento. Para esta vista se suelen usar los diagramas UML de secuencia, comunicaciones o de actividad.

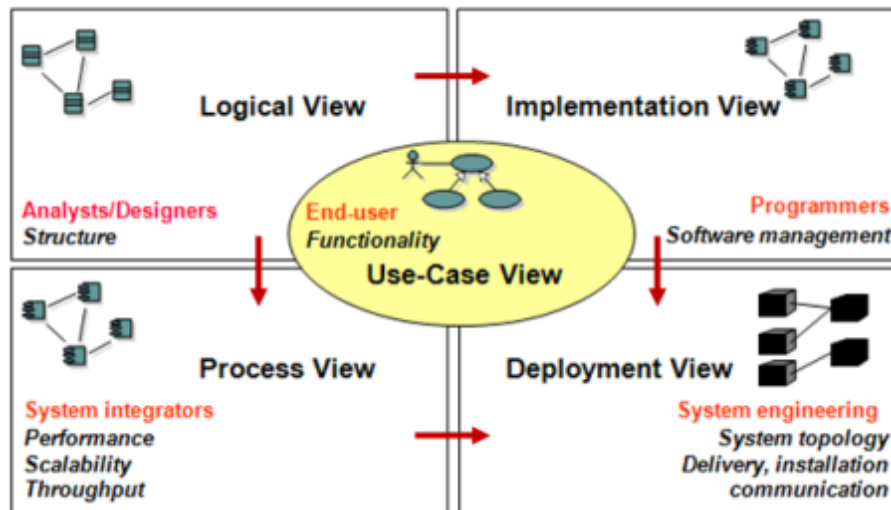


Figura 4.2: Modelo 4+1 de vista arquitectural (Kruchten 1995)

- **Vista de desarrollo:** En esta vista se enfoca el sistema bajo el punto de vista de la gestión de código, presentando los varios paquetes, capas o componentes en que se divide naturalmente el código, así como las relaciones entre ellos. Se suelen usar los diagramas UML de componente o de paquetes.
- **Vista física:** Muestra las particularidades y necesidades de la arquitectura bajo el punto de vista de un ingeniero de sistemas, teniendo en cuenta requerimientos no-funcionales como la disponibilidad (*availability*), la tolerancia a fallos (*reliability*), el rendimiento o la escalabilidad. Identifica elementos y componentes del sistema y los mapea a varios nodos físicos (procesadores), teniendo en cuenta las conexiones entre ellos. Para esta vista se suele usar el diagrama UML de despliegue (*deployment diagram*).
- **Escenarios:** Los escenarios son la abstracción de los requerimientos funcionales más relevantes. Describen secuencias de interacciones entre los actores y el sistema. Esta vista es redundante con las otras (y por ello es la vista “+1”), pero tiene dos propósitos principales: como vector para descubrir los varios elementos arquitecturales que poblarán las otras vistas en la fase de diseño; y para ilustrar y validar la arquitectura

una vez el diseño está completado, tanto sobre el papel como en la fase de pruebas.

4.3.2. Casos de uso

A continuación se describen los casos de uso (o escenarios) principales, resumidos en la figura 4.3.

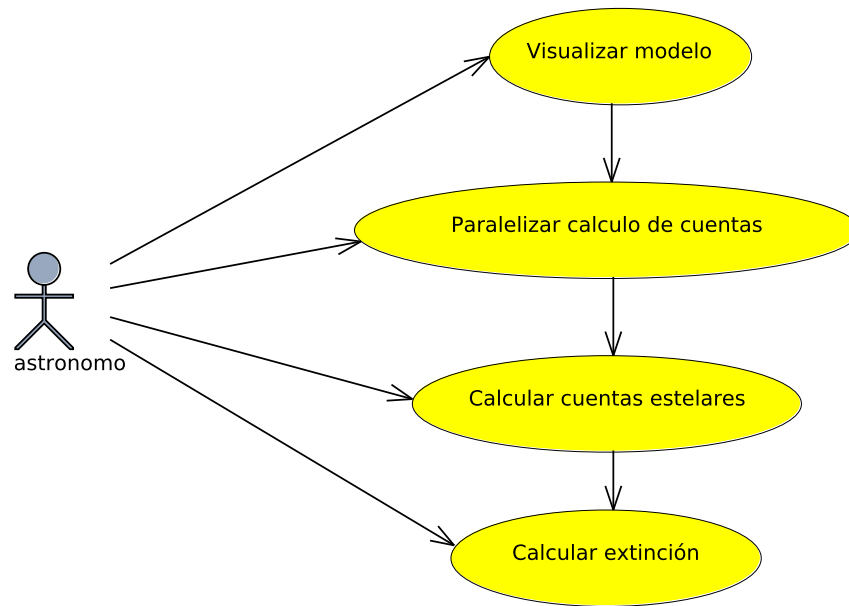


Figura 4.3: Diagrama de casos de uso principales

Calcular cuentas estelares en una coordenada celeste

Caso de uso	Calcular cuentas estelares
Descripción	Este caso de uso permite calcular la predicción del número de cuentas estelares que se obtendrán en una determinada coordenada celeste, observando con cierto filtro y hasta la magnitud límite dada.
Trigger	El usuario o componente del sistema ejecuta la función
Flujo primario	<ol style="list-style-type: none"> 1. El usuario configura las coordenadas galácticas de interés 2. El usuario configura la banda observacional 3. El usuario configura la magnitud aparente límite 4. El usuario configura el campo aparente 5. El usuario configura el tamaño del paso en la línea de visión 6. El sistema transforma de coordenadas galácticas esféricas a coordenadas cartesianas galactocéntricas 7. El sistema calcula la extinción en cada punto de la línea de visión 8. Para cada punto de la línea de visión, el sistema calcula la magnitud absoluta a la que corresponde la magnitud aparente configurada 9. El sistema calcula la densidad de cada componente en los puntos de la línea de visión 10. El sistema integra la función de luminosidad en cada punto de la línea de visión 11. El sistema calcula el integrando de la ecuación fundamental de la estadística estelar (3.1), en cada punto de la línea de visión 12. Usando los cálculos anteriores, el sistema integra la expresión (3.1) y devuelve el resultado
Condición de final correcto	La expresión (3.1) se ha integrado y el sistema ha devuelto el resultado
Condición de final incorrecto	La expresión (3.1) no se ha podido integrar por algunos de los siguientes motivos:

	<ul style="list-style-type: none"> ■ Las coordenadas galácticas introducidas son incorrectas ■ La banda observacional introducida no está soportada ■ No se pueden leer los ficheros de las funciones de luminosidad
Flujo en caso de error	El sistema debe avisar al usuario y abortar el cálculo

Calcular extinción

Caso de uso	Calcular extinción
Descripción	Este caso de uso permite calcular la extinción causada por el medio interestelar desde el Sol hasta cierto punto de la galaxia, usando el modelo de Drimmel et al. (2003)
Trigger	El usuario o componente del sistema ejecuta la función
Flujo primario	<ol style="list-style-type: none"> 1. El usuario configura la coordenada galáctica de interés, y el radio heliocéntrico 2. El usuario configura la banda observacional 3. El usuario configura si se debe realizar una corrección con los datos de DIRBE/COBE 4. El sistema realiza una interpolación trilineal de los datos de extinción del disco, para la coordenada dada y para la banda V 5. El sistema realiza una interpolación trilineal de los datos de extinción de los brazos espirales, para la coordenada dada y para la banda V 6. El sistema realiza una interpolación trilineal de los datos de extinción del brazo local, para la coordenada dada y para la banda V

	<p>7. A partir de la extinción en la banda V, el sistema calcula la extinción en la banda seleccionada por el usuario</p> <p>8. El sistema aplica los factores de corrección calculados en el flujo alternativo</p> <p>9. El sistema devuelve el valor de extinción calculado</p>
Flujos alternativos	<p>En el paso 8 del flujo principal, si el usuario no ha seleccionado el uso de los datos de DIRBE, se toma un valor de 1,0 para los factores de corrección.</p> <p>En el paso 8 del flujo principal, si el usuario ha seleccionado el uso de los datos de DIRBE, se aplica el siguiente algoritmo para el cálculo de los factores de corrección:</p> <ol style="list-style-type: none"> 1. El sistema transforma las coordenadas galácticas a coordenadas eclípticas 2. El sistema calcula el pixel de DIRBE que corresponde a las coordenadas, usando una proyección esférica cuadrilateralizada (White and Stemwedel 1992) 3. El sistema obtiene el valor de los factores de corrección para este pixel
Condición de final correcto	El sistema calcula y devuelve el valor de la extinción causada por la columna de medio interestelar entre el Sol y la coordenada galáctica dada
Condición de final incorrecto	<p>La extinción no se ha podido calcular, por algunos de los siguientes motivos</p> <ul style="list-style-type: none"> ■ Las coordenadas galácticas introducidas son incorrectas ■ La banda observacional introducida no está soportada ■ No se pueden leer los ficheros de datos de extinción ■ No se puede leer el fichero de datos de DIRBE
Flujo en caso de error	El sistema debe avisar al usuario y abortar el cálculo

Calcular cuentas estelares de un rango de coordenadas celestes en paralelo

Caso de uso	Paralelizar cálculo de cuentas en un rango de coordenadas
Descripción	Este caso de uso permite calcular la predicción de cuentas estelares en un rango de longitudes galácticas, latitudes galácticas, o ambas
Trigger	El usuario o componente del sistema ejecuta la función
Flujo primario	<ol style="list-style-type: none"> 1. El usuario configura una longitud galáctica, o un rango de longitudes 2. El usuario configura una latitud galáctica, o un rango de latitudes 3. El usuario configura la banda observacional 4. El usuario configura la magnitud aparente límite 5. El usuario configura el campo aparente 6. A partir de las coordenadas o rangos de coordenadas, introducidos el sistema calcula el <i>grid</i> correspondiente 7. El sistema realiza una partición apropiada del <i>grid</i> de coordenadas y la distribuye entre los varios nodos de cálculo disponibles 8. El sistema distribuye los parámetros del modelo entre los varios nodos de cálculo disponibles 9. Cada nodo de cálculo ejecuta concurrentemente el caso de uso “Calcular cuentas” para cada coordenada recibida 10. El sistema recoge los resultados producidos en cada nodo de cálculo, los ordena según la organización del <i>grid</i> de coordenadas inicial y devuelve el resultado
Condición de final correcto	El sistema calcula y devuelve el valor de la expresión (3.1) para los rangos de coordenadas demandados

Condición de final incorrecto	<p>No se ha podido realizar el cálculo por alguno de los siguientes motivos:</p> <ul style="list-style-type: none"> ▪ El rango de coordenadas galácticas introducido es incorrecto ▪ La banda observacional introducida no está soportada ▪ Se produce un fallo de sistema que impide la comunicación con los nodos de cálculo
Flujo en caso de error	El sistema debe avisar al usuario y abortar el cálculo

Visualizar predicción de cuentas y compararla con catálogos observacionales

Caso de uso	Visualizar modelo
Descripción	Este caso de uso permite cargar un catálogo y comparar los resultados observacionales con la predicción aportada por el modelo
Trigger	El usuario ejecuta la aplicación gráfica
Flujo primario	<ol style="list-style-type: none"> 1. El sistema carga los datos del catálogo 2. El usuario configura la magnitud aparente límite 3. El usuario configura el rango de longitudes galácticas de interés 4. El usuario configura la latitud galáctica de interés 5. Opcionalmente, el usuario puede cambiar los parámetros del modelo 6. Opcionalmente, el usuario puede cargar los parámetros del modelo a partir de fichero 7. Opcionalmente, el usuario puede guardar los parámetros modificados del modelo a fichero 8. El usuario requiere al sistema la visualización de la predicción del modelo y del catálogo

	9. El sistema ejecuta el caso de uso “ Paralelizar cálculo de cuentas en un rango de coordenadas ” 10. El sistema crea una gráfica para visualizar el resultado del cálculo del modelo y el contenido del catálogo 11. Opcionalmente el usuario puede hacer <i>zoom</i> en regiones de interés de la gráfica 12. Opcionalmente el usuario puede guardar la gráfica en un fichero de imagen
Condición de final correcto	El crea y visualiza una gráfica interactiva con los datos del modelo y del catálogo
Condición de final incorrecto	No se ha podido crear la gráfica por alguno de los siguientes motivos: <ul style="list-style-type: none"> ▪ El fichero del catálogo no está disponible ▪ Se produce algún error en la ejecución del caso de uso “Paralelizar cálculo de cuentas en un rango de coordenadas”
Flujo en caso de error	El sistema debe avisar al usuario con un mensaje de error adecuado

4.3.3. Vista de desarrollo

El modelo Galiac no se ha implementado de forma monolítica sino que se ha descompuesto en varios paquetes, agrupando código según la funcionalidad de alto nivel que proporciona. El paquete principal se denomina *galiac*, y de él cuelgan los siguientes módulos:

- **core:** Este módulo contiene las clases principales de la aplicación: la clase *Model*, que implementa el modelo en sí; la jerarquía representada por *GalacticComponent* y sus clases derivadas, que implementan las funciones de densidad de cada componente de la Galaxia; la clase *ModelParameters*, que almacena los parámetros del modelo; y la clase *Coordinates*, para el manejo de coordenadas galácticas.
- **extinction:** Agrupa el código relacionado con el cálculo de la extinción provocada por el material del medio interestelar. Contiene una

clase base, que define la interfaz básica, y las clases derivadas que implementan modelos de extinción específicos. En la actualidad, el único modelo soportado es el de Drimmel.

- **parallel:** Contiene el código que implementa la capacidad de paralelizar el cálculo del modelo. La clase base *galiac.parallel.ModCalc* define la interfaz, y de ella se derivan las que implementan el paralelismo con diversas estrategias. Los dos mecanismos soportados son SMP y MPI.
- **catalog:** En *galiac.catalog* tenemos el código relacionado con el manejo de catálogos. Como en los módulos anteriores, contiene una clase base que define la interfaz y funcionalidad básica, y otras que derivan de ella para implementar las características específicas de cada catálogo. En la presente versión, se soporta el catálogo 2MASS mediante la clase *TwoMASS*.
- **gui:** Este módulo contiene el código de la aplicación gráfica: la clase *GaliacGUI* y sus clases auxiliares, además de una sección *__main__* que permite que éste módulo se pueda ejecutar como un script Python.

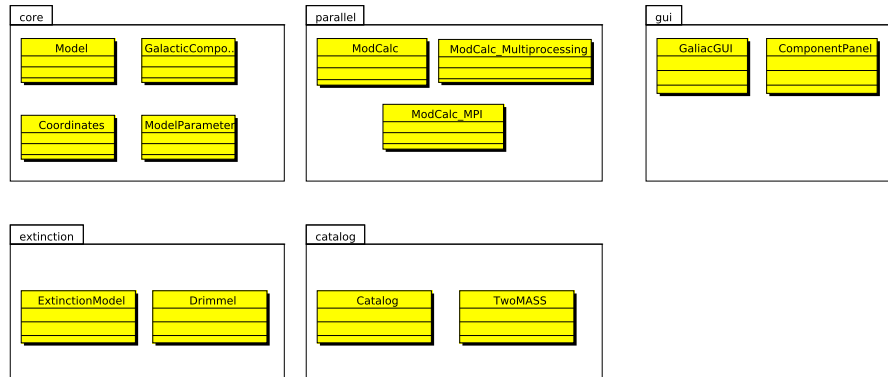


Figura 4.4: Organización del paquete *galiac*

4.3.4. Vista lógica

A continuación se muestra la vista lógica de la arquitectura del modelo *Galiac*, es decir, las entidades lógicas (clases) que se identifican en el dominio

y los escenarios, así como las relaciones estáticas entre ellos. Por motivos de espacio, dividiremos esta vista en un apartado para cada módulo descrito en la vista de desarrollo.

Paquete *core*

La figura 4.5 muestra el diagrama de clases del módulo *galiac.core*. La clase principal es la ***Model***, que es la encargada del cálculo de la predicción de cuentas estelares. Para ello, la clase implementa algunas funciones con cálculos que son globales al dominio (como el paso de magnitudes aparentes a absolutas, o la integración global del resultado), pero en general delega a las otras clases los cálculos específicos, consiguiendo alta cohesión (ver sección 4.2.2). Además, siguiendo también el patrón *creator* de GRASP, es la responsable de crear los objetos del resto de clases del módulo. La clase *Model* expone al usuario todo el interfaz necesario para el cálculo y gestión del modelo (asignación de parámetros, etc), con lo cual éste se desacopla del resto de clases.

La función principal de la clase *Model* es *star_counts*, cuya signatura es:

```
def star_counts (self, l, b, w, m, dr=0.1, in_radians=False):
```

donde l es la longitud galáctica, b es la latitud galáctica, w es el tamaño angular del campo de visión, m la magnitud aparente, dr el tamaño del paso de integración en kiloparsecs (con un valor por defecto de 100 parsecs), y *in_radians* controla si los parámetros angulares están expresados en grados decimales o en radianes (grados cuadrados o estereorradianes en el caso de w).

La clase *Model* está relacionada por composición con la clase ***GalacticComponent***. Ésta es la clase base de la jerarquía de clases responsables del cálculo de la densidad estelar de cada componente de la Galaxia. Las clases que heredan de *GalacticComponent* son: *ThinDisk*, *ThickDisk*, *Bulge*, *Bar*, *Halo* y *SpiralArms*. La clase base *GalacticComponent* declara el método que usará *Model* para solicitar el cálculo de la densidad:

```
def density (self, coords):
```

a la cual simplemente se le pasa un objeto de tipo *Coordinates*. Esta función devuelve un valor nulo para la clase base, y las clases derivadas la reimplementan para devolver los resultados de las expresiones listadas en la sección 3.4. Además, la clase base proporciona el método *set_passband*, que selecciona

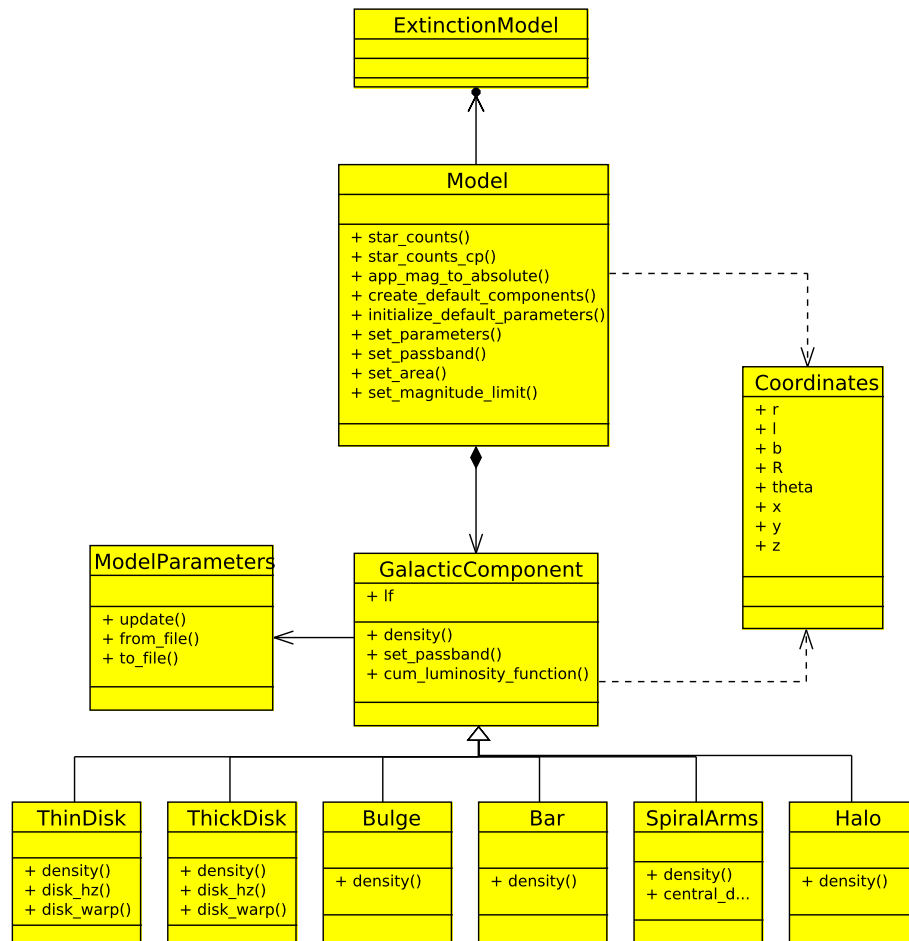


Figura 4.5: Diagrama de clases del paquete *galpac.core*

la función de luminosidad adecuada según la banda observacional requerida, y también exporta como propiedad la función de luminosidad acumulada:

```
@property
def cum_luminosity_function(self):
```

La clase *ModelParameters* implementa la funcionalidad e interfaz necesarias para la gestión de los parámetros del modelo. Proporciona métodos para cargar y guardar los parámetros a fichero de disco (métodos *from_file* y *to_file*) y para actualizarlos desde un diccionario Python (método *update*). La clase *ModelParameters* ofrece como atributos los parámetros del modelo, que son dependientes de la formulación de cada componente galáctico, pero sin embargo según las recomendaciones de los patrones GRASP sería indeseable que la clase estuviera acoplada a la jerarquía de *GalacticComponent*. Se ha conseguido proporcionar esta funcionalidad y evitar al mismo tiempo el acoplamiento aprovechando la característica dinámica de Python e implementando la clase mediante el patrón *SimpleNamespace* (ver sección 4.4.4).

Finalmente, la clase *Coordinates* implementa la funcionalidad para almacenar y transformar coordenadas. Su constructor recibe coordenadas galácticas esféricas y una lista de distancias heliocéntricas:

```
def __init__(self, l, b, r, in_radians=False):
```

Posteriormente exporta una serie de propiedades que permiten obtener las mismas coordenadas, pero transformadas a coordenadas cartesianas galactocéntricas (x, y, z) o a coordenadas cilíndricas galactocéntricas (R, θ) .

La transformación de coordenadas no se realiza en el constructor, sino que se aplica la técnica de inicialización retardada para calcularla sólo en el caso que se necesite (ver sección 4.4.3). Una vez calculada alguna transformación, el resultado se guarda en cache para no repetirla.

Paquete *extinction*

En el paquete *galiac.extinction* tenemos sólo dos clases: *ExtinctionModel* y *Drimmel*. La primera de ellas es la clase base de la jerarquía de modelos de extinción. Declara una interfaz genérica para un modelo de extinción, que es la que usa la clase *galiac.core.Model* a fin de abstraer a ésta del modelo de extinción usado realmente:

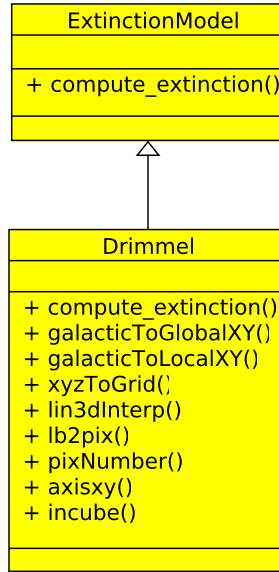


Figura 4.6: Diagrama de clases del paquete *galiax.extinction*

```
def compute_extinction (self, coords, filt='K', params={}):
```

Por otra parte, la clase *Drimmel* implementa el modelo de extinción galáctico descrito en la sección 3.2. Esta clase carga los datos del modelo de Drimmel et al. (2003), y realiza una interpolación trilineal para determinar el valor de la extinción en la coordenada galáctica requerida. Además, opcionalmente se puede aplicar un factor de corrección al valor anterior usando los datos en infrarrojo lejano de DIRBE/COBE.

Paquete *parallel*

En el paquete *galiax.parallel*, cuyo diagrama de clases se puede ver en la figura 4.7, tenemos la implementación de la paralelización del modelo. Igual que en el paquete anterior, existe una clase base, *ModCalc*, que declara la interfaz de alto nivel, y luego hay dos clases que proporcionan la implementación según dos estrategias de paralelización específicas: *ModCalc_Multiprocessing* y *ModCalc_MPI*. Además, la clase base contiene una instancia de *core.Model*, a la cual delega el cálculo en sí del modelo. El método principal de la interfaz de *ModCalc* para el cálculo paralelo del modelo

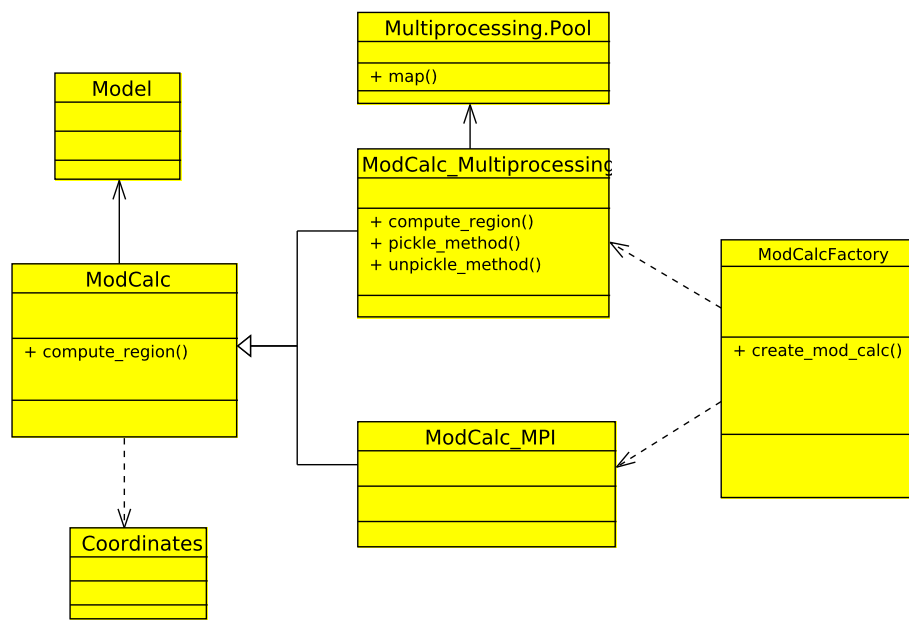


Figura 4.7: Diagrama de clases del paquete *galiac.parallel*

es:

```
def compute_region (self, l, b, area, mag, band):
```

donde ahora l y b no son una coordenada individual sino un vector de coordenadas, y devuelve el resultado del cálculo del modelo para todas ellas.

La clase *ModCalc_Multiprocessing* implementa el cálculo paralelo del modelo según la estrategia SMP (*Symmetric Multiprocessing*). Como se comenta en la sección 4.5.1, en la implementación estándar de Python existe un mecanismo llamado *Global Interlock* que imposibilita el uso de threads para la paralelización de un cálculo intensivo entre los varios núcleos de una máquina. Para evitar este problema, la clase *ModCalc_Multiprocessing* utiliza la librería *Multiprocessing*, que usa procesos (no threads) para conseguir el paralelismo. En concreto, las instancias de *ModCalc_Multiprocessing* tienen una instancia de la clase *multiprocessing.Pool*, que implementa un pool de threads.

Paquete *gui*

El paquete *galiac.gui*, cuyo diagrama de clases se muestra en la figura 4.8, contiene el código relacionado con la interfaz gráfica. Python ofrece numerosas opciones para la implementación de interfaces gráficas, como por ejemplo *WxPython*, *PyGUI*, *PyQT*, *PyKDE*, o *TkInter*. Se decidió usar ésta última por ser la “estándar” de Python, y por lo tanto la mejor soportada por las varias distribuciones e implementaciones del lenguaje.

La clase principal es *GaliacGUI*, que hereda de la clase de *Tkinter ttk.Frame* y es, por lo tanto, la ventana raíz del árbol de ventanas y widgets de la aplicación. En el constructor de *GaliacGUI* se crean los paneles y elementos gráficos generales, y de una instancia de *ComponentPanel* para cada componente del modelo galáctico. Además, la clase *GaliacGUI* contiene una instancia de *ModCalc_Multiprocessing* para ejecutar el cálculo del modelo en paralelo cuando lo solicita el usuario, así como una instancia de *Catalog* para la consulta de las cuentas estelares observacionales. Una vez calculado el modelo y consultado el catálogo, la gráfica correspondiente se crea mediante la librería *matplotlib*.

La clase *ComponentPanel* implementa de forma genérica un panel de configuración de los parámetros de un componente dado del modelo galáctico. La creación y gestión de los editores de propiedades se hace de forma dinámica, de modo que esta clase no necesita estar acoplada a los parámetros específicos de cada componente del modelo. El constructor recibe por parámetros

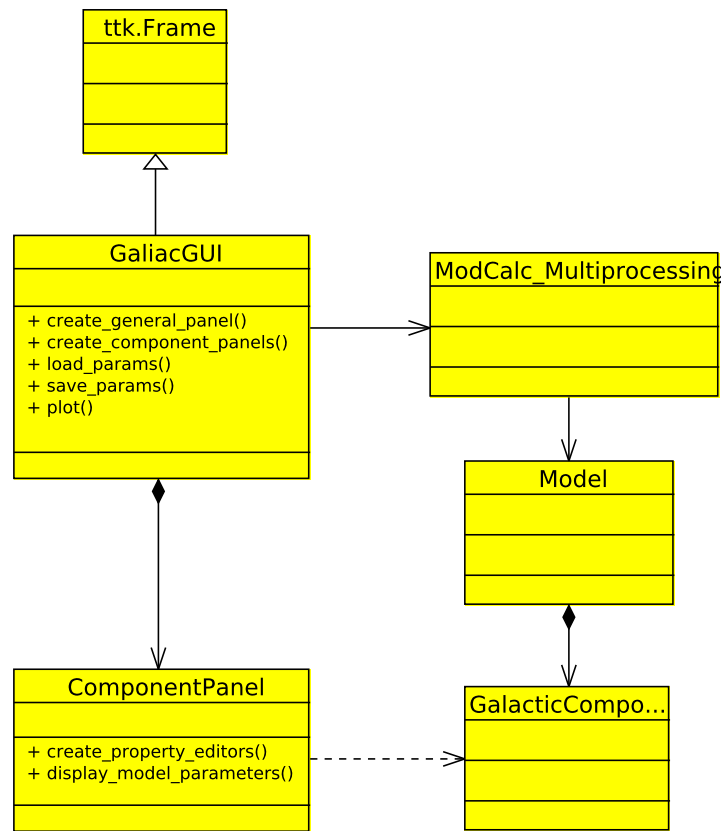


Figura 4.8: Diagrama de clases del paquete *galiec.gui*

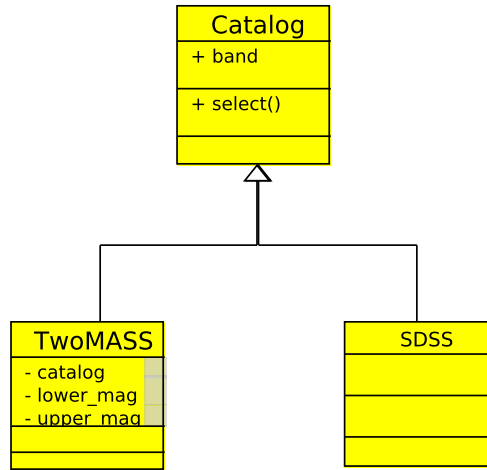


Figura 4.9: Diagrama de clases del paquete *galactic.catalog*

un objeto derivado de *GalacticComponent*, al cual le solicita el listado de nombres de sus parámetros de configuración. Entonces crea dinámicamente una etiqueta y una caja de edición para cada parámetro, manejando correctamente los varios tipos de datos de cada propiedad mediante el uso de las capacidades de introspección de Python.

Paquete *catalog*

En la figura 4.9 se puede encontrar el diagrama de clases del paquete *catalog*. Como en casos anteriores, tenemos una clase base abstracta que define la interfaz, *Catalog*, y una jerarquía de clases derivadas que implementan el código específico para acceder a cada tipo concreto de catálogo. Por el momento, el catálogo soportado es el de infrarrojo cercano 2MASS, y la clase que lo implementa es *TwoMass*.

El método principal de las clases que heredan de *Catalog* es el de selección de una región del cielo:

```
def select (self, l=None, b=None, mag=14):
```

donde *l* y *b* pueden ser *None*, un valor en concreto o una tupla de dos elementos. Esto permite, de una forma sencilla, especificar coordenadas concretas o rangos de ellas. Si el parámetro introducido es un valor concreto, se toma

como condición en la búsqueda del catálogo que la coordenada en cuestión debe ser igual al valor dado. Si es una tupla, el primer valor indica el inicio del rango de coordenadas que se seleccionará, y el segundo el final. Si es *None*, entonces cualquier valor para esta coordenada cumplirá con la condición de selección. Por ejemplo, si tenemos $l = (10, 40)$ y $b = 3$, se seleccionarán todas las entradas del catálogo cuya latitud galáctica sea 3, y cuya longitud esté entre 10 y 40; o si tenemos $l = 10$ y $b = \text{None}$, se seleccionarán todas las entradas cuya longitud galáctica sea 10, sin importar el valor de la latitud

4.3.5. Vista de proceso

A continuación se describirá la parte dinámica de la arquitectura, es decir, la secuencia de mensajes que intercambian las clases descritas en la vista lógica para cumplir con los requerimientos de los casos de uso.

Operación *star_counts*

En la figura 4.10 podemos observar el diagrama de secuencia de la operación *galiac.core.Model.star_counts*, que implementa el caso de uso “**Calcular cuentas estelares**”. La clase que coordina toda la operación es *core.Model*, y va delegando en las otras para cada paso específico. En primer lugar, *Model* crea el objeto de coordenadas a partir de los datos proporcionados por el usuario. Posteriormente realiza la transformación de magnitudes aparentes a magnitudes absolutas, para lo cual delega en *extinction.ExtinctionModel* el cálculo de la extinción. Posteriormente itera a través de todos los componentes galácticos activos del modelo, solicitando polimórficamente el cálculo de la densidad (que cada clase derivada de *GalacticComponent* tiene reimplementada), y la integración de la función de luminosidad para cada magnitud absoluta. Finalmente, calcula el integrando de la expresión 3.1) y realiza la integración numérica para obtener el resultado.

Operación *compute_region*

La figura 4.11 muestra el diagrama de secuencia de la operación *parallel.ModCalc.compute_region*, que implementa mediante SMP el caso de uso “**Paralelizar cálculo de cuentas en un rango de coordenadas**”. Vemos que se mantiene el criterio de alta cohesión, puesto que cada componente está focalizado en sus responsabilidades: *ModCalc_Multiprocessing* conoce

los detalles sobre cómo paralelizar un cálculo, pero delega en *Model* para el cálculo en sí (y por otra parte, éste último no está acoplado a los detalles de paralelización, puesto que no entra dentro de sus responsabilidades).

Operación *plot*

Finalmente, en la figura 4.12 se encuentra el diagrama de secuencia de la operación que implementa el caso de uso “**Visualizar modelo**”. Esta operación es, realmente, el *handler* que la clase *GaliacGUI* le configura al componente gráfico del botón *Plot*.

Como en casos anteriores, para efectuar esta operación la clase *GaliacGUI* hace las funciones de coordinación, delegando en las clases apropiadas cada paso específico: los parámetros del modelo se interrogan a las varias instancias de *ComponentPanel*, las cuentas estelares observacionales se consiguen consultando a la clase derivada de *Catalog*, la predicción de cuentas estelares del modelo se calculan en paralelo mediante una instancia de la clase *ModCalc_Multiprocessing*, y finalmente se visualiza la gráfica usando la librería *matplotlib*.

4.3.6. Vista física

En esta sección mostramos la vista física de la arquitectura, es decir, el modo en que se la puede desplegar en un sistema hardware. Como el sistema implementado en este TFC se puede usar de múltiples formas (por ejemplo, como librería, como aplicación gráfica, paralelizando con MPI, como página web, etc) realmente no tendríamos un conjunto cerrado de diagramas de despliegue que representen todas las posibilidades. Por ello, con fines ilustrativos detallaremos dos posibles despliegues: como aplicación gráfica (mediante paralelismo SMP) o usándose como librería desde una aplicación de ajuste y paralelizando a gran escala mediante MPI.

En la figura 4.13 tenemos el diagrama de despliegue del caso de usar el modelo *Galiac* como aplicación gráfica. Vemos que todo el sistema se ejecuta en una sola máquina (normalmente, un PC de escritorio) pero cuyo procesador tiene varios núcleos. En uno de los núcleos se ejecuta la aplicación gráfica implementada en la clase *GaliacGUI*, la cual mediante el uso de la clase *ModCalc_Multiprocessing* envía peticiones de cálculo en paralelo a los modelos que tenemos ejecutándose en el resto de núcleos del procesador. La comunicación entre procesos se realiza usando memoria compartida.

sd: star_counts

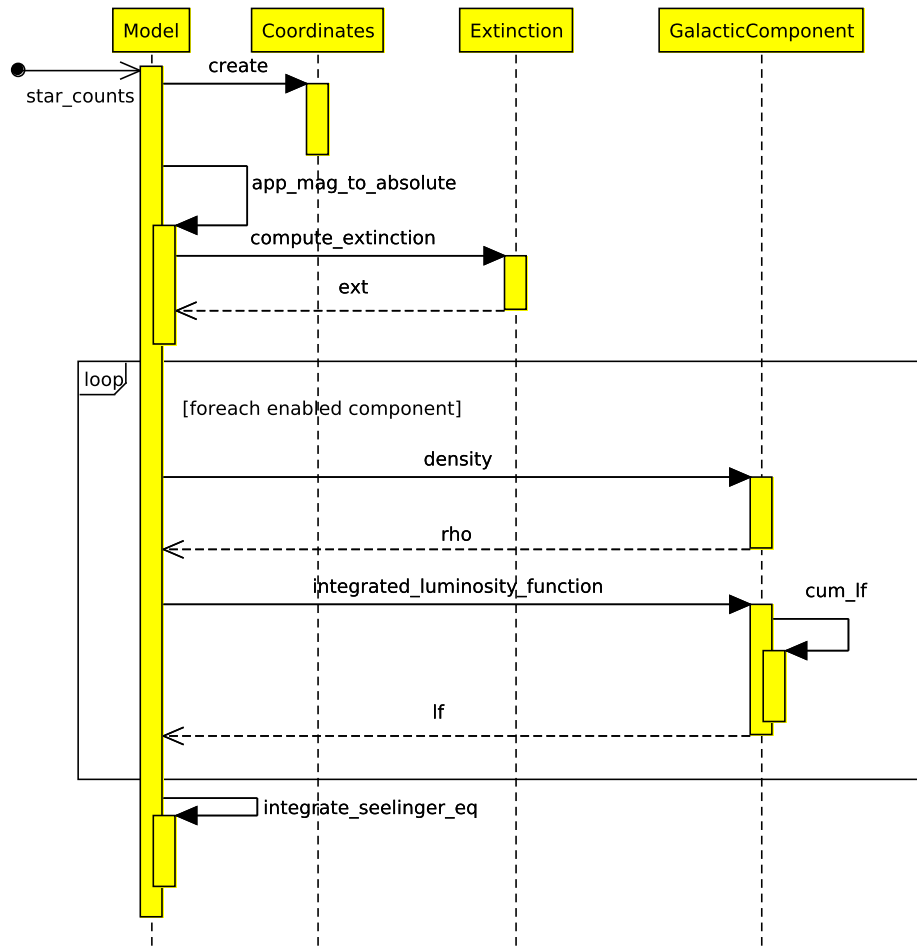


Figura 4.10: Diagrama de secuencia de la operación *star_counts*

sd: compute parallel model

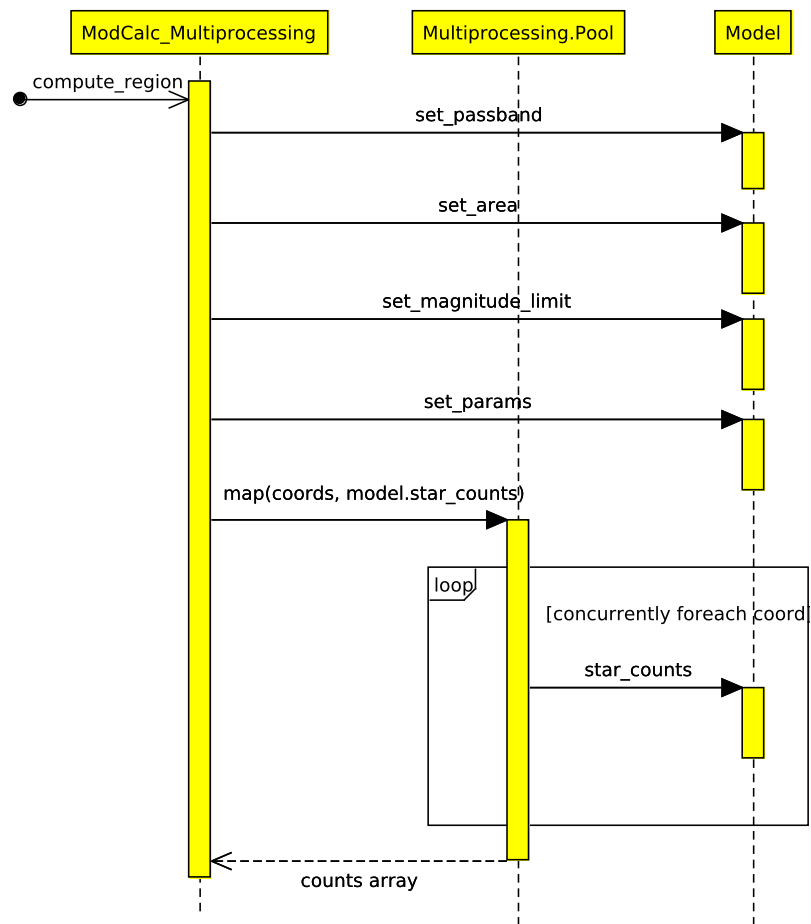


Figura 4.11: Diagrama de secuencia de la operación *compute_region*

sd: plot

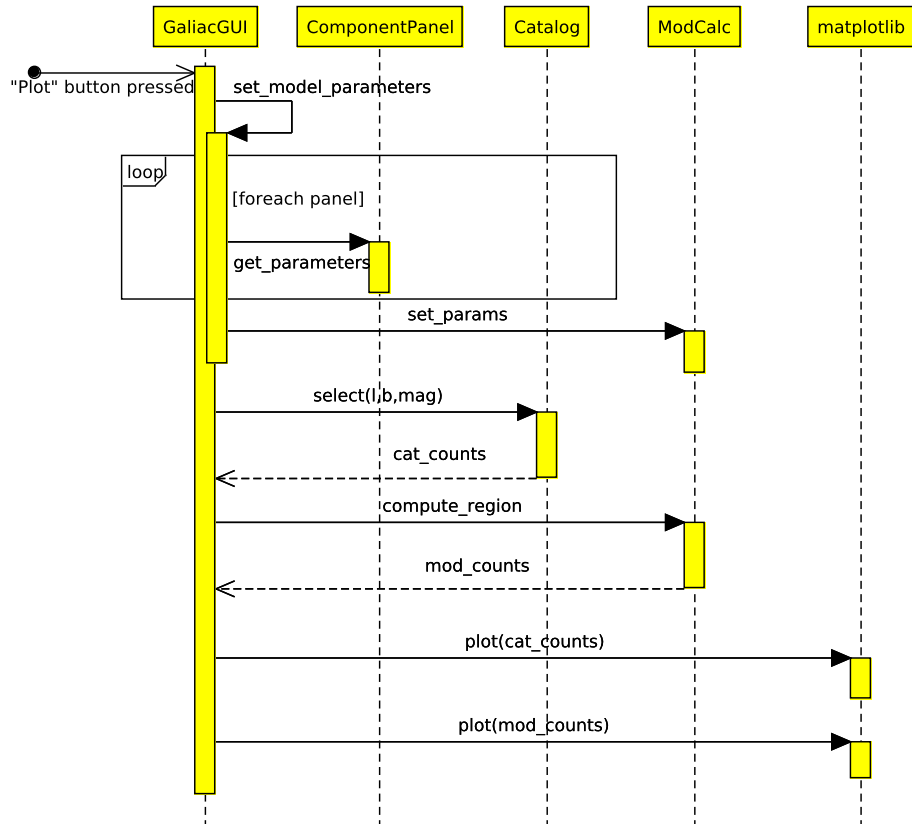


Figura 4.12: Diagrama de secuencia de la operación *plot*

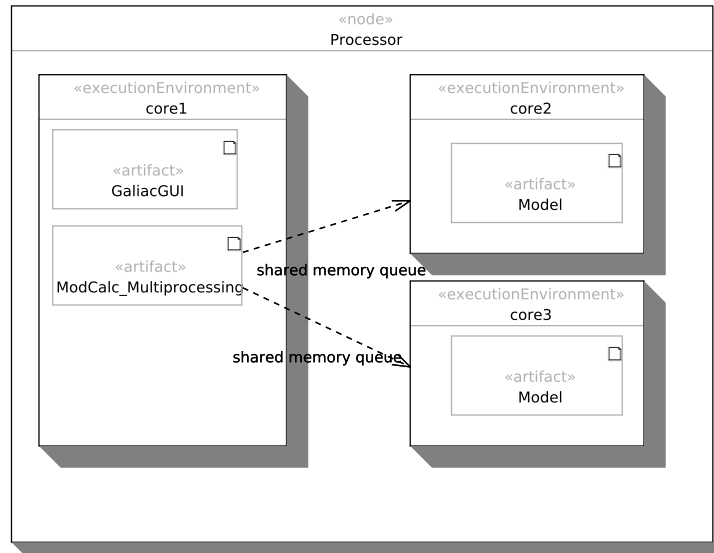


Figura 4.13: Diagrama de despliegue usando paralelismo SMP

Por otra parte, en la figura 4.14 tenemos el despliegue para el caso de usar paralelismo MPI. En este caso el sistema se despliega a lo largo de varios nodos de cómputo, que pueden formar parte de, por ejemplo, un cluster tipo Beowulf (Becker et al. 1995) o un cluster de alto rendimiento. En uno de los nodos, que actúa como *master*, se ejecuta la aplicación interesada en los cálculos del modelo (por ejemplo, una aplicación de ajuste por mínimos cuadrados), la cual usa la clase *ModCalc_MPI* para comunicarse con los nodos de cálculo distribuidos. Esta clase usa el protocolo MPI a través de la red para propagar al resto de nodos los parámetros del modelo y las coordenadas solicitadas, y estos efectúan el cálculo y devuelven el resultado también por MPI.

4.4. Detalles de implementación

Dos de los criterios que se han aplicado en más intensidad a la hora de tomar decisiones de diseño o de implementación en el modelo han sido la eficiencia computacional y la mantenibilidad futura del código. En los siguientes apartados se dan detalles de algunas partes del código donde se

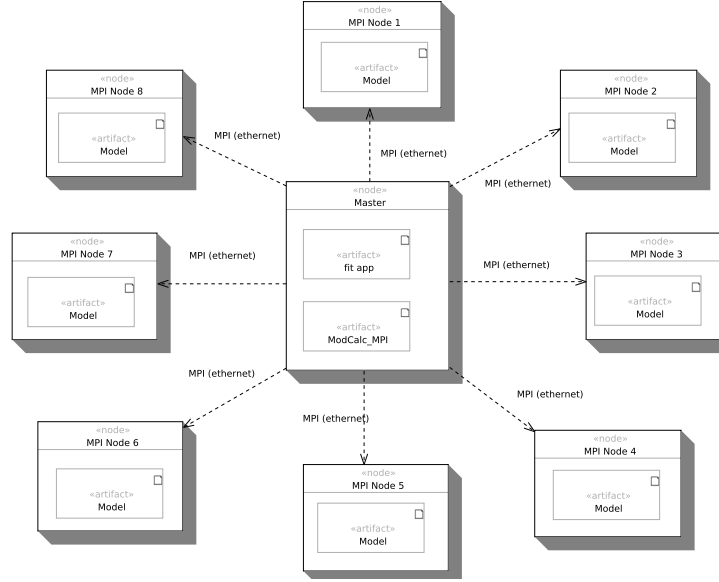


Figura 4.14: Diagrama de despliegue usando paralelismo MPI

han aplicado estos principios.

4.4.1. Implementación de la integración de la función de luminosidad

Como se puede ver en la expresión (3.1), el integrando de la ecuación fundamental de la estadística estelar depende de $\Phi(M(m, \vec{r}))$. Por otro lado, hay que recordar que Φ es la función de distribución acumulada de la función de luminosidad ϕ (ver ecuación (3.3)). Como ésta depende de la magnitud absoluta límite M , que a su vez depende \vec{r} , entonces en cada paso de la integración numérica de la ecuación principal hay que realizar otra integración numérica para hallar Φ .

Este paso es crítico puesto que se repite muchas veces y, por lo tanto, tiene un impacto muy significativo en el rendimiento. La implementación *naïve*, en la cual realizamos el cálculo entero de la integración en cada iteración multiplica el coste temporal del algoritmo. Sin embargo, analizando mejor el problema nos podemos dar cuenta que, realmente, hay una gran parte del cálculo que se repite. Por lo tanto, podemos evitar esta repetición y aumentar

la eficiencia.

Tenemos una función de luminosidad definida de forma numérica y discretizada en un *array* cuyas celdas representan un conjunto ordenado de magnitudes, $M = \{M_0, M_1, \dots, M_n\}$, y cuyos valores expresan el valor de la función de luminosidad en cada una de ellas, $\phi(M_i)$. Queremos hacer una integración numérica entre $-\infty$ y cierta magnitud M_x . En general, M_x no coincidirá con los bordes de ninguna celda, sino que estará en el interior de alguna ella, $M_i \leq M_x < M_{i+1}$. El cálculo de $\Phi(M_x)$ lo podemos dividir en

$$\Phi(M_x) = \int_{-\infty}^{M_x} \phi(M') dM' = \int_{-\infty}^{M_i} \phi(M') dM' + \int_{M_i}^{M_x} \phi(M') dM' \quad (4.1)$$

Pero resulta claro que, para cada celda M_i el primer sumando de la expresión anterior será siempre el mismo, y no depende de la posición galáctica \vec{r} que estamos considerando en concreto. Por lo tanto, al inicio construimos un segundo *array* que contiene en cada celda i el resultado de integrar la función de luminosidad hasta M_i . Este cálculo se realiza una sola vez y se guarda en memoria para ser reutilizado en el resto de iteraciones. Entonces, para el cálculo de $\Phi(M_x)$ sólo debemos calcular el segundo sumando de la expresión (4.1), que es mucho menos costoso que el cálculo entero, y añadirlo al valor precalculado de $\Phi(M_i)$. En términos de coste computacional, hemos reducido el orden de magnitud del coste de $O(n)$ a $O(1)$.

El siguiente código Python es el que implementa este algoritmo:

```
def integrated_luminosity_function(self, component, Mlim):
    '''
    For each absolute magnitude in the Mlim array, it
    computes the integral of the luminosity function of
    the component, from an absolute magnitude of -infinity
    to Mlim
    '''

    intlf = component.cum_luminosity_function
    lf = component.luminosity_function
    ind = np.searchsorted(lf[0], Mlim)
    ind_before_interval = np.where(Mlim <= lf[0][0])
    ind_after_interval = np.where(ind >= lf[0].size)
    ind[ind_after_interval] = -1
    res = np.where(ind >= 2, intlf[ind-2], 0.0)
```

```

res += (Mlim-lf[0][ind-1])*(lf[1][ind-1]
      +0.5*(lf[1][ind]-lf[1][ind-1])
      *(Mlim-lf[0][ind-1])/(lf[0][ind]-lf[0][ind-1]))
res[ind_before_interval] = 0.0
res[ind_after_interval] = 1.0

return res

```

Como se puede ver, el cálculo se ha podido vectorizar, de modo que no se computa Φ para cada magnitud absoluta, sino que se hace de forma conjunta para todas las magnitudes absolutas límite que se corresponden con cada uno de los pasos de la integral en la línea de visión. De este modo, el cálculo es aún más eficiente, puesto que evitamos bucles (que en Python son costoso). La dificultad principal ha recaído en el hecho que los elementos que hay que consultar del array de la función de luminosidad son diferentes para cada M_x , pero esto se ha solventado mediante las capacidades de Python y Numpy de construir un subarray indexando a otro con una lista de índices. Por ejemplo, la sentencia

```
ind = np.searchsorted(lf[0], Mlim)
```

construye un array de índices (*ind*), que apunta cada uno a la celda de la función de luminosidad cuya magnitud es inmediatamente superior a cada uno de los valores del *array* M_{lim} .

4.4.2. Implementación de las funciones de densidad

Como se ha explicado anteriormente, las clases derivadas de *GalacticComponent* implementan las funciones de densidad descritas en la sección 3.4. Así pues, las clases *ThinDisk* y *ThickDisk* implementan las ecuaciones de (3.17), la clase *Bulge* implementa la ecuación (3.21), la clase *Bar* implementa (3.23), la clase *SpiralArms* implementa la ecuación (3.27) y, finalmente, la clase *Halo* es la responsable de la implementación de la ecuación (3.28).

Como ejemplos de algunas de ellas, a continuación se lista el código de las implementaciones de las funciones de densidad del disco fino (3.17) y de los brazos espirales (3.27). La del disco fino pertenece a la clase *ThinDisk*, y su interés radica principalmente en que es la que genera la mayor parte de las cuentas estelares observadas:


```

def density (self, coords):
    '''
    This function returns the stellar disk density in the
    given coordinates
    @param coords: The Coordinates object with the galactic
                    locations where the density shall be computed
    '''
    p = self.params
    R = coords.R
    z = coords.z

    #compute scale height and warp
    hz = self.disk_hz (coords)
    zw = self.disk_warp (coords)

    #compute density
    return (p.A
            *(p.hz_thin_sun/hz)
            *np.exp(p.Rsun/p.hr_thin + p.hr_hole/p.Rsun)
            *np.exp(-R/p.hr_thin-p.hr_hole/R)
            *np.exp(-abs(z-zw)/hz))

def disk_hz (self, coords):
    '''
    Returns the scale height of the disc in the
    given coordinates
    @param coords: The Coordinates object with the galactic
                    locations where the scale height
                    must be computed
    '''
    p = self.params
    R = coords.R
    return (p.hz_thin_sun
            *(1+p.hz_hole_lin*(R-p.Rflare)
              +p.hz_hole_sq*(R-p.Rflare)**2))

def disk_warp (self, coords):
    ''' Compute the value of the warp '''

```

```

p = self.params
theta = coords.theta
return (p.Cwarp
        *((R*1e3)**p.Ewarp)
        *np.sin(theta-p.theta_warp)+p.Zsun)*1e-3

```

Por otra parte, la implementación de la función de densidad de los brazos espirales reside en la clase *SpiralArms*. Como se puede observar, es bastante más compleja que la del disco fino, puesto que la densidad en los brazos está definida en función de la coordenada de la espiral y de la distancia a ella. Dada una coordenada cilíndrica galactocéntrica (R, θ) primero debemos encontrar el punto de la espiral (3.25) más cercano a ella, solucionando la ecuación (3.26). Posteriormente, si la distancia no supera el ancho del brazo, calculamos la densidad a partir de la densidad central del brazo en aquella zona:

```

def density (self, coords):
    '''
    This function returns the arms stellar density in the
    given coordinates
    @param coords: The Coordinates object with the galactic
                    locations where the density shall be computed
    '''
    p = self.params
    z = coords.z
    R = coords.R
    theta = coords.theta

    #parameters of the spiral
    b = math.tan(p.armPitch)
    a = p.armInitialRadius*math.exp(-b*p.armInitialAngle)
    a2 = p.armInitialRadius*math.exp(-b*(p.armInitialAngle+pi))

    # calculate the floating point approximation for n
    n1 = (np.log(R/a)/b - theta)/(2.0*pi)
    n2 = (np.log(R/a2)/b - theta)/(2.0*pi)

    # find the two possible radii for the closest point
    upper_r1 = a * np.exp(b * (theta + 2.0*pi*np.ceil(n1)))

```

```

lower_r1 = a * np.exp(b * (theta + 2.0*pi*np.floor(n1)))
upper_r2 = a2 * np.exp(b * (theta + 2.0*pi*np.ceil(n2)))
lower_r2 = a2 * np.exp(b * (theta + 2.0*pi*np.floor(n2)))

newR = np.where(np.abs(upper_r1-R)<p.sigmaArm, upper_r1, 0)
ind = np.abs(R - lower_r1)<p.sigmaArm
newR[ind] = lower_r1[ind]
ind = np.abs(upper_r2 - R)<p.sigmaArm
newR[ind] = upper_r2[ind]
ind = np.abs(R - lower_r2)<p.sigmaArm
newR[ind] = lower_r2[ind]
newR[R>17.0] = 0.0

ind = newR>p.armInitialRadius
armPointsR = newR[ind]
armPointsZ = z[ind]
armPointsTheta = theta[ind]
rhoArmPoints = self.spiralArmsCentralDensity (p,
                                                armPointsR,
                                                armPointsTheta,
                                                armPointsZ)

rhoarm = np.zeros(R.size)
rhoarm[ind] = rhoArmPoints
return rhoarm

```

4.4.3. Inicialización retardada

En el modelo hay numerosos cálculos o transformaciones cuyo resultado se necesita en múltiples sitios. Unos ejemplos de ello son la transformación de coordenadas heliocéntricas a galactocéntricas, o el cálculo de las funciones de luminosidad acumuladas que se han mencionado en el apartado anterior. Si estos cálculos se repitieran en cada ocasión donde son necesarios, estaríamos ocasionando una pérdida de eficiencia. Sin embargo, si hacemos un precálculo inicial de todas las posibles transformaciones también estaríamos incurriendo en un código ineficiente, puesto que es posible que algunas de ellas no sean necesarias para el cálculo que está solicitando el usuario en cada ocasión.

Una estrategia para resolver el problema es la **inicialización retardada**. Mediante esta técnica, sólo realizamos el cálculo la primera vez que se

necesita, y a partir de este momento queda guardado en memoria para su posterior uso repetitivo. El método concreto que hemos usado es definir como propiedades aquellos valores susceptibles de esta optimización y asignar a la variable subyacente un valor inicial de *None*. El código de la propiedad verifica primero el valor de esta variable: si es *None* realiza el cálculo y lo almacena, pero si no lo es simplemente devuelve el valor precalculado.

Por ejemplo, este es el código de la clase *Coordinates* que devuelve el valor de la componente *x* de la transformación de coordenadas heliocéntricas a galactocéntricas:

```
@property
def x (self):
    ''' Cartesian galactocentric X coordinates '''
    if self._x is None:
        self._x = (self.Rsun
                    - self._r*np.cos(self._b)*np.cos(self._l))
    return self._x
```

Y su uso sería:

```
coord = Coordinates(90, 10, np.linspace(0.0, 30.0, 1000))
print coord.x #here array 'x' is computed for first time
print coord.x #here array 'x' is not recomputed, but used
               #from cached value
```

4.4.4. Implementación de la clase *ModelParameters*

La clase *ModelParameters* debe almacenar los parámetros del modelo, además de proporcionar funcionalidad para guardarlos a disco y recuperarlos posteriormente. El problema radica en que cada componente galáctico tiene un conjunto de parámetros que dependen de su definición e implementación, y además éstos pueden estar compartidos entre varios componentes. No se quería acoplar la clase *ModelParameters* al conjunto de parámetros concretos que usamos en la definición actual del modelo, puesto que ésta puede cambiar en el futuro. La solución fue usar las características dinámicas de Python, que permiten añadir o modificar atributos de un objeto en tiempo de ejecución. En concreto, usamos el patrón de *SimpleNamespace*, según el cual modificamos directamente la variable miembro especial *_dict_*:

```

def __init__(self, **kwargs):
    self.update(kwargs)

def update (self, new_parms):
    self.__dict__.update (new_parms)

```

Cada componente define un mapa con sus parámetros por defecto, que son añadidos luego dinámicamente al objeto de *ModelParameters*. Posteriormente cada componente puede acceder a dichos parámetros como si fueran miembros normales de la clase:

```

p = ModelParameters()
p.update ({'bulgeAxialRatio': np.array([1.0, 0.49, 0.40])})
print p.bulgeAxialRatio

```

La funcionalidad para serializar los parámetros a disco se ha obtenido mediante el uso de las funciones *load* y *dump* del paquete *pickle*.

4.4.5. Implementación de la selección en el catálogo

Como se ha comentado en la sección de la vista lógica, las clases derivadas de *Catalog* permiten fijar varios criterios de búsqueda. Esto se consigue usando las capacidades de indexación de Numpy, con el siguiente código Python:

```

ind = np.repeat (True, self.catalog.shape[0])
#select using longitude condition, if it is present
if l is not None:
    #check if it is an interval
    try:
        linit = l[0]
        lend = l[1]
        ind &= ((self.catalog[:,0] >= linit)
               & (self.catalog[:,0] <= lend))
    except (TypeError, IndexError):
        #it is not an interval, but a specific longitude
        ind &= self.catalog[:,0] == l

#select using latitude condition, if it is present

```

```

if b is not None:
    #check if it is an interval
    try:
        binit = b[0]
        bend = b[1]
        ind &= ((self.catalog[:,1] >= binit)
                & (self.catalog[:,1] <= bend))
    except (TypeError, IndexError):
        #it is not an interval, but a specific longitude
        ind &= self.catalog[:,1] == b

```

4.5. Paralelización

Un algoritmo como el del modelo de cuentas estelares requiere una gran potencia de cómputo, a causa del número elevado de cálculos que requiere la integración numérica de la ecuación (3.1) si se quiere realizar para rangos extensos de (l, b) . Un natural para disminuir el tiempo de cómputo es abordar técnicas de computación paralela.

El algoritmo del modelo es muy apto para su paralelización, puesto que los parámetros de entrada involucrados en la obtención de las cuentas estelares para cada coordenada (l, b) son independientes de los resultados del cálculo en otras coordenadas. Si disponemos de N procesadores y deseamos calcular las cuentas estelares en M coordenadas, una estrategia clásica para paralelizar este tipo de algoritmos consiste en hacer una partición de la entrada del problema (es decir, el conjunto de coordenadas galácticas sobre las cuales evaluamos la ecuación (3.1) en N subconjuntos de M/N puntos, y asignar cada uno de ellos a un procesador diferente. Finalmente, se recogen los subconjuntos del resultado que ha producido cada procesador, y se ensamblan en la solución final. Los detalles de cómo repartir las entradas y obtener los resultados de cada procesador varían en función de la técnica concreta de paralelismo que se escoja.

En el caso del presente Trabajo de Fin de Carrera, se ha dado soporte a dos arquitecturas de cómputo paralelo: por una parte a sistemas multiprocesador SMP (*symmetric multiprocessing*); y por la otra a sistemas paralelos en *cluster* mediante el uso de MPI.

4.5.1. SMP

Un sistema SMP consiste en una computadora dotada de múltiples procesadores que acceden a una misma memoria RAM vía un bus compartido. En un sistema SMP cada procesador ejecuta una tarea en paralelo de forma independiente del resto de procesadores, pero siendo posible compartir memoria entre ellos. El sistema operativo puede cambiar de forma sencilla las tareas asignadas a cada procesador para balancear la carga entre ellos y optimizar el sistema, puesto que todos los procesadores tienen un acceso simétrico e uniforme a la memoria donde residen los datos. Habitualmente, el sistema operativo encapsula la complejidad de la planificación de los procesos, y presenta el sistema al usuario como si fuera una única máquina.

En la actualidad, la mayor parte de las CPU presentes en el mercado son SMP (los famosos “multicore”), y hay una gran cantidad de sistemas operativos que pueden gestionar máquinas SMP (entre los cuales se encuentran también los sistemas más extendidos actualmente, como Linux o las últimas versiones de Windows). Por lo tanto, dotar al modelo Galiac de la capacidad de ejecución paralela en sistemas SMP implica que automáticamente se mejorará el rendimiento del mismo en la mayor parte del hardware existente, incluso en PC estándares de nivel bajo/medio.

En Linux, que es el sistema de referencia en el que se ejecuta Galiac, se puede obtener ventaja de un sistema SMP mediante dos mecanismos: ya sea con *multithreading* (con espacios de memoria compartidos) o bien con múltiples procesos (con espacios de memoria independientes). El estado compartido, inherente al primer mecanismo, hace que sea más sencillo programar la concurrencia en algoritmos como el presente. Sin embargo, desafortunadamente el intérprete CPython (que es la implementación estándar de Python) está implementado con una estrategia GIL (*Global Interpreter Lock*), que causa que el intérprete sólo puede ejecutar un hilo (“*thread*”) a la vez, con lo cual desaparecen todas las ventajas del paralelismo. Por lo tanto, la concurrencia para sistemas SMP en Galiac se ha implementado mediante múltiples procesos, que sí que aseguran el paralelismo.

Para ello, se ha usado el paquete **multiprocessing**, que forma parte de la librería estándar de Python. Este paquete ofrece varias clases que permiten la creación de procesos, la comunicación inter-proceso, la compartición de memoria, etc. En el presente Trabajo de Final de Carrera se ha usado la clase `multiprocessing.Pool`, que permite crear un conjunto de procesos que quedan en espera de recibir trabajo. En nuestro caso, creamos tantos

Num proc.	Tiempo 64800 puntos	Tiempo/ punto	S_p	E_p
serie	215,69 s	3,32 ms	–	–
1	217,15 s	3,35 ms	0,99	0,99
2	113,09 s	1,74 ms	1,90	0,95
3	84,61 s	1,30 ms	2,54	0,84
4	63,64 s	0,98 ms	3,38	0,84
5	64,44 s	0,99 ms	3,34	0,66
6	60,58 s	0,93 ms	3,56	0,59
7	56,54 s	0,87 ms	3,81	0,54
8	52,55 s	0,81 ms	4,10	0,51

Tabla 4.5: Rendimiento del modelo con paralelización SMP, según el número de procesadores usado

procesos como procesadores existen en la máquina, para obtener la totalidad de la capacidad del hardware y sin entrar en problemas causados por cambios de contexto demasiado frecuentes en un mismo procesador. Posteriormente, se usa la función `map` de la clase `Pool` para hacer la partición de la lista de coordenadas y asignar subconjuntos de coordenadas a cada uno de los procesos.

Para comprobar el aumento del rendimiento al usar varios procesadores, se han realizado pruebas en una máquina con CPU Intel Core i7, que dispone de 8 procesadores a 2.8 GHz. Se han medido los tiempos necesarios para el cómputo de las cuentas estelares en todo el cielo empleando un número diferente de procesadores (hasta el límite de la máquina), y se ha comparado con el tiempo requerido para la versión en serie, sin usar computación paralela. En concreto el cálculo del modelo se ha realizado sobre un conjunto de 64800 líneas de visión (360×180), y se ha medido el tiempo total, el tiempo por línea de visión, el factor de aceleración (“*speedup*”) y la eficiencia. Estos dos últimos términos son una medida de la ventaja del algoritmo paralelo respecto al algoritmo serie, $S_p = T_1/T_p$ y $E_p = T_1/(pT_p)$, donde T_1 es el tiempo de ejecución sin paralelismo, T_p es el tiempo de ejecución con p procesadores en paralelo, S_p es el factor de aceleración y E_p la eficiencia (0 implica ninguna ventaja de la versión paralela respecto a la serie, y 1 significa un paralelismo

perfecto).

En la tabla 4.5 se encuentran los tiempos obtenidos. Podemos ver que usando 8 procesadores el tiempo que se demora **el cálculo del modelo por cada línea de visión es de tan sólo 0.8 milisegundos**, en comparación a los 3.3 milisegundos que se requería en la versión no paralela (o respecto a los 3 segundos que requiere el modelo SKY). Esto implica un factor de aceleración de 4, e implica un ahorro de tiempo considerable si se aplica el modelo sobre conjuntos grandes de coordenadas. La eficiencia, por su parte, empieza a caer más pronunciadamente a partir de 5 procesadores, lo cual está causado porque en un modelo SMP el acceso a memoria se realiza mediante un *bus* compartido, y por lo tanto todos los procesadores compiten por este recurso.

4.5.2. MPI

La ventaja de un sistema SMP es que es de muy fácil acceso, y su programación es también relativamente sencilla. Sin embargo, tiene una limitación importante en el hecho que no se puede escalar, puesto que no es posible añadir más procesadores a la CPU que los que ya lleva de fábrica.

Otra estrategia de procesamiento paralelo, que no presenta el problema anterior, es la computación en *cluster*. En este modelo, varias máquinas independientes se unen en un mismo conjunto de forma que se puedan gestionar como una sola y puedan colaborar en la ejecución de tareas. Normalmente estos nodos se conectan entre ellos mediante ethernet (o soluciones de mayor prestaciones, como myrinet), y no tienen acceso a memoria compartida. El cluster escala fácilmente añadiendo más nodos, y como está formado por varios elementos que individualmente no es necesario que tengan capacidades de computación muy altas, es una de las soluciones con mejor cociente entre precio y rendimiento, y por lo tanto una de las soluciones preferidas en super-computación.

Al no disponer de memoria compartida, los procesadores deben intercambiar datos mediante otras estrategias. La más extendida de ellas en clusters de súper-computación es la de “paso de mensajes”. Uno de los sistemas de paso de mensajes más usado es MPI, que está disponible para varias arquitecturas y lenguajes de programación, como C, Fortran, C++, Java, Python, etc. Hay varias implementaciones del protocolo MPI, siendo las más conocidas la de MPICH y la de LAM/MPI. En nuestro caso se ha elegido la segunda.

En una aplicación paralela que use MPI, los diferentes procesos realizan sus cálculos de forma independiente sin compartir memoria, pero con la capa-

cidad de sincronizarse o enviarse datos entre ellos, ya sea en modo *broadcast* o punto a punto. MPI define una serie de conceptos y estándares para los roles, tipos de datos y operaciones de sincronización e intercambio de mensajes. Para estas últimas, MPI soporta comunicaciones síncronas y asíncronas, operaciones de paso de mensajes de punto a punto, operaciones de repartición de datos de uno a muchos, o de muchos a uno, etc. El contenido de los mensajes intercambiados es arbitrario y definido por la aplicación, lo cual aporta una gran flexibilidad pero también es más complejo de implementar.

La estrategia adoptada para paralelizar Galiac mediante MPI es de tipo maestro-esclavo: hay un proceso que adopta el rol de maestro y varios procesos que adoptan el rol de esclavo (idealmente, tantos esclavos como procesadores disponibles). El proceso maestro es el encargado de la gestión del resto de procesos, así como de leer configuraciones y datos auxiliares, y de realizar la partición del conjunto de coordenadas de la entrada. Cuando un proceso esclavo es iniciado se queda en espera de recibir un mensaje por parte del maestro. Luego, el maestro envía a cada esclavo uno de los subconjuntos de coordenadas y el vector con los parámetros del modelo. Posteriormente cada esclavo calcula de forma independiente las cuentas estelares para el subconjunto de la entrada que ha recibido, y envía el resultado mediante un mensaje MPI al maestro. Finalmente, éste último coloca cada solución parcial en el lugar correcto del vector de la solución final.

Para probar el modelo con paralelización por MPI se ha montado un cluster LAM/MPI con 6 PC estándares. Cada uno de ellos tiene una CPU Intel Core i7 2.8 GHz con 8 procesadores, con lo cual el cluster ofrece 48 procesadores (aunque al final hubo que prescindir de 3 nodos por motivos ajenos, quedando entonces en un total de 3 nodos y 24 procesadores). Se hicieron varias pruebas, modificando el número de nodos del cluster y el número de procesadores por nodo.

En la tabla 4.6 se encuentran los resultados obtenidos del rendimiento del sistema. Observamos que con la configuración con el máximo de procesadores (24) **obtenemos un tiempo para el cálculo del modelo para una línea de visión de tan sólo 0.38 milisegundos.**

Adicionalmente, los ajustes que se detallan en la sección 5.2.2 se han realizado todos con este cluster. Como para cada ajuste hay que realizar varias iteraciones de cálculo del modelo, la mejora en el tiempo de proceso respecto a una solución sin paralelizar ha sido muy notable.

Num nodos	Proc./ nodo	Proc. totales	Tiempo 64800 puntos	Tiempo/ punto	S_p	E_p
serie	1	1	215,69 s	3,32 ms	–	–
2	2	4	63,06 s	0,97 ms	3,42	0,85
2	4	8	46,53 s	0,71 ms	4,63	0,57
2	6	12	42,26 s	0,65 ms	5,10	0,42
2	8	16	29,39 s	0,45 ms	7,33	0,45
3	2	6	56,09 s	0,86 ms	3,84	0,64
3	4	12	31,56 s	0,48 ms	6,83	0,56
3	6	18	28,55 s	0,44 ms	7,55	0,41
3	8	24	24,65 s	0,38 ms	8,74	0,36

Tabla 4.6: Rendimiento del modelo con paralelización MPI, según el número de nodos y de procesadores por nodo usados

4.6. Uso como librería

Es posible acceder al modelo Galiac desde otra aplicación, cargándolo como una librería externa. La opción más sencilla es usarlo desde otro código en *Python*, en cuyo caso lo único necesario es que el modelo esté en los directorios donde *Python* busca los módulos, y añadir la siguiente sentencia al inicio del programa:

```
import galiac.core as gc
```

A partir de este momento podemos acceder a la clase *Model*, bajo el namespace *gc*. Por ejemplo, el siguiente código crea una instancia del modelo con los parámetros por defecto, y realiza un cálculo de predicción de cuentas estelares en la longitud galáctica del anticentro, en el plano del disco, en un área de 1 grado cuadrado, con el filtro K y hasta magnitud 14:

```
import galiac.core as gc

#create the galactic model
mod = gc.Model()

#configure filter
```

```
mod.set_passband('K')

#compute stellar counts
counts = mod.star_counts (self, 180, 0, 1, 14)
```

Si se dispone de aplicaciones en otros lenguajes de programación también es posible acceder al modelo, puesto que hay varias implementaciones que permiten ejecutar el intérprete de *Python* desde otros lenguajes. Algunos ejemplos son el sistema *Jython* para ejecutarlo desde *Java*, la librería *Boost/Python* para ejecutarlo desde C++, etc.

4.7. Interfaz web

Uno de los requerimientos del proyecto era que el modelo debía estar disponible en una versión vía página web, puesto que de este modo es accesible a un mayor número de investigadores con objetivos más heterogéneos.

Para ello se ha procedido a crear una página web que permite el cálculo de la predicción del modelo de las cuentas estelares en el rango de coordenadas galácticas especificado por el usuario. La interfaz es sencilla, basada en un formulario en el cual el usuario puede introducir los parámetros del modelo y de la consulta. Cada una de las casillas correspondiente a un parámetro contiene el valor por defecto de dicho parámetro, de forma que sólo es necesario introducir aquellos valores que se quieran cambiar. Cuando el usuario ha terminado de introducir los parámetros y valida el formulario, se dispara el proceso de cálculo en el servidor y una vez terminado se ofrece para su descarga.

Para la ejecución del modelo desde un servidor web se ha escrito un CGI en Python que importa la implementación del modelo como librería. El CGI recoge y valida los parámetros que ha introducido el usuario, y ejecuta el modelo mediante la estrategia de paralelización SMP.

Para su acceso abierto desde Internet, se ha procedido a configurar un dominio y un servidor web. El modelo está disponible en la *url* <http://galiac.molgo.com>. En la figura 4.15 se puede encontrar una captura de pantalla del mismo.

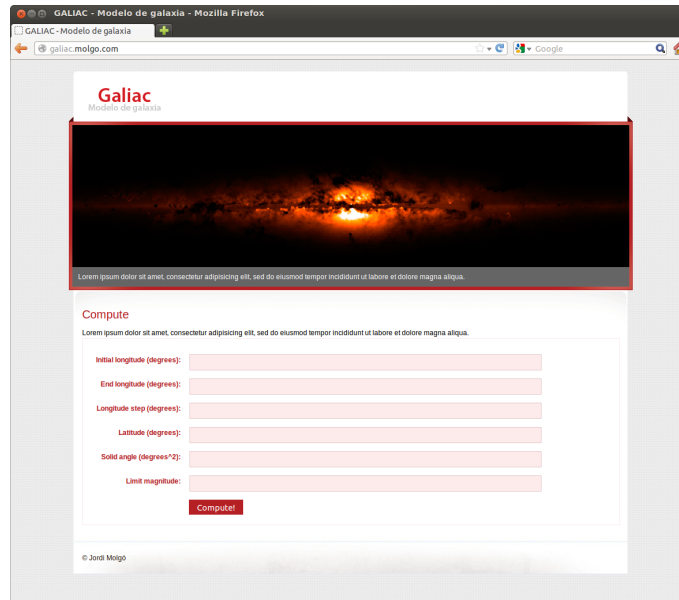


Figura 4.15: Interfaz web para el modelo Galiac (<http://galiac.molgo.com>)

4.8. Aplicaciones de usuario

Se han creado tres aplicaciones de usuario para facilitar el uso del modelo: dos aplicaciones gráficas que permiten hacer gráficas de los resultados del modelo y navegar por las funciones de densidad de los componentes, y una aplicación en modo texto para realizar los ajustes de la sección 5.2.2.

galiac-gui: Interfaz del modelo

Esta aplicación permite trabajar con el modelo de una forma más interactiva. Como se ha comentado en la sección de arquitectura, la implementación de la aplicación gráfica la proporciona la clase *GaliacGUI*, del módulo *galiac.gui*.

La interfaz gráfica permite cambiar los parámetros del modelo, guardar los parámetros o cargarlos desde un fichero de configuración, ejecutar el modelo sobre un rango de coordenadas galácticas determinado y mostrar gráficamente los resultados comparando con los de 2MASS. El cálculo del modelo se realiza usando la versión paralelizada con SMP, y para la implementación de la parte gráfica se usaron las librerías **Tkinter** y **matplotlib**.

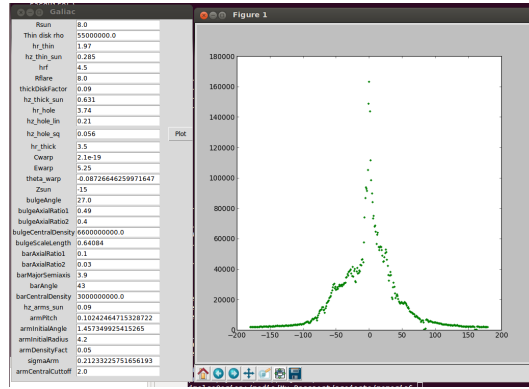


Figura 4.16: Aplicación galiac-gui: interfaz gráfica del modelo

galiac-fit: Ajuste de parámetros

La funcionalidad de la aplicación **galiac-fit** es la de realizar ajustes de los parámetros del modelo a las cuentas estelares que provienen de datos observacionales. Esta tercera aplicación, a diferencia de las dos anteriores, no se ejecuta en modo gráfico sino en modo texto y su uso es algo más complicado.

Entre todas las implementaciones disponibles y verificadas del algoritmo de Levenverg-Marquardt se ha elegido la proporcionada por el paquete de Python *Scipy*, que a su vez utiliza la librería MINPACK.

El usuario elige, mediante un fichero de configuración, los parámetros que se van a ajustar, la zona del cielo que va a usar, los valores del resto de parámetros del modelo, y el fichero que se va a usar como catálogo.

El cálculo de los residuales para la obtención de χ^2 se ha implementado usando la versión paralelizada con el *cluster* de MPI, de forma que los ajustes se ejecutan eficientemente en tiempos razonables. Hay un nodo maestro, que es el que divide el conjunto de puntos total de entrada en varios conjuntos más pequeños (uno por cada nodo esclavo) y dirige el ajuste, de forma que cada vez que necesita calcular los residuales solicita a los nodos esclavo que calculen paralelamente el modelo para las coordenadas que se les ha sido asignadas. Posteriormente, los residuales se pesan con la desviación estándar correspondiente a una distribución poissoniana, y se ejecuta el algoritmo de ajuste, obteniendo así los nuevos valores para los parámetros.

Capítulo 5

Resultados

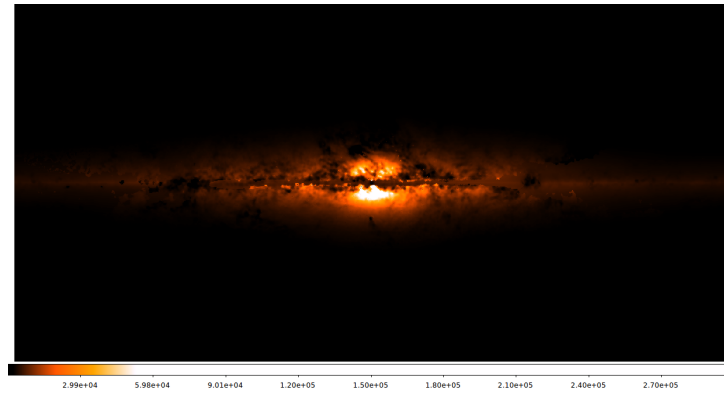


Figura 5.1: Mapa de cuentas estelares de la Vía Láctea obtenida con el modelo Galiac en la banda V

En esta sección se detallan los resultados obtenidos con el modelo. Uno de los intereses principales en el modelo es realizar ajustes de los parámetros estructurales a los datos, a fin de mejorar nuestro conocimiento sobre ellos. Como trabajo preliminar, primero se ha realizado un estudio con el objetivo de encontrar las mejores zonas del cielo para el ajuste de cada componente. Posteriormente, se ha procedido a realizar los ajustes de los parámetros del disco, el bulbo, la barra y los brazos espirales. Finalmente, se comparan los resultados con los datos observacionales.

5.1. Datos observacionales

Entre todos los cartografiados disponibles, en el presente Trabajo de Fin de Carrera hemos trabajado principalmente con el catálogo de fuentes puntuales (*Point Source Catalog*) de 2MASS. Se eligió este cartografiado por su completitud, y porque el infrarrojo está menos afectado de extinción. Los ajustes se han realizado con los datos de 2MASS, y posteriormente se han usado también las cuentas estelares procedentes del *Sloan Digital Sky Survey* para comparar los resultados del modelo en las longitudes de onda del visible.

5.1.1. 2MASS

2MASS (*Two Micron All Sky Survey* Skrutskie et al. (2006)), es un cartografiado infrarrojo de todo el cielo (con una cobertura del 99,998 %) en las bandas J, H y K_s, con capacidad para detectar fuentes puntuales más brillantes que 1 mJy con una relación señal a ruido mayor que 10 y una precisión astrométrica de 0.5". La sensibilidad del catálogo llega hasta las magnitudes límite de 15.8 para la banda J, 15.1 para H y 14.3 en K_s, en las zonas donde la confusión de fuentes no es muy elevada. Para las fuentes con una SNR mayor que 20, los valores fotométricos tienen un margen de error del 5 %.

Para la realización de 2MASS se han empleado dos telescopios de 1.3 metros, uno situado en el hemisferio Norte (Arizona) y otro en el hemisferio Sur (Chile). Ambos están equipados con una cámara de 3 canales para la observación simultánea en JHK, y cada canal está compuesto por un detector de HgCdTe de 256×256 elementos.

El telescopio norte empezó la operación en 1997, mientras que el telescopio sur lo hizo en 1998. En mayo de 1999 se liberó la "*First Incremental Release*", con un 6 % del cielo. Posteriormente, en marzo del 2000 apareció la "*Second Incremental Release*", con un 47 % del cielo, y finalmente en marzo de 2003 apareció la "*All Sky Release*", con el 99,998 % del cielo en ambos hemisferios. Los productos de datos que el proyecto 2MASS ha puesto a disposición de la comunidad astronómica son el "*Digital Sky Atlas*", que contiene los cuatro millones de imágenes del cartografiado en las tres bandas; el "*Extended Source Catalog*" (XSC), con las posiciones y magnitudes de más de un millón de galaxias y nebulosas; y el "*Point Source Catalog*" (PSC), con las posiciones y magnitudes de unos 300 millones de estrellas. Este último producto es el que nosotros hemos utilizado para el ajuste y comparación con la salida del modelo.

5.1.2. Sloan Digital Sky Survey

El SDSS, “*Sloan Digital Sky Survey*”, es un cartografiado en longitudes de onda del visible, y llega hasta las magnitudes límite 22.5, 22.5, 22.5, 22.0 y 21.5 en las bandas u, g, r, i, z respectivamente. Los datos se adquieren con un telescopio de 2.5 metros, situado en el Apache Point Observatory, de Nuevo México.

El proyecto del SDSS ha experimentado tres fases: el SDSS-I, que se ejecutó entre el año 2000 y el 2005; el SDSS-II, entre el 2005 y el 2008; y el SDSS-III, que empezó en 2008 y terminará en 2014. En el presente trabajo hemos utilizado última versión de los datos que se ha liberado es la DR9, de julio del 2012.

La cobertura final del cielo que tendrá el SDSS será del 30 % aproximadamente. El proyecto original estaba más orientado a estudios extragalácticos, por lo cual la zona de cobertura se centró principalmente alrededor de los polos galácticos y se excluyó la zona del plano. Sin embargo, a partir del SDSS-II empezó el subproyecto SEGUE (“*Sloan Extension for Galactic Understanding and Exploration*”), para el estudio de la estructura de la Vía Láctea, principalmente de la parte externa. En el SDSS-III el proyecto SEGUE se renueva con SEGE-II, y se complementa con el proyecto APOGEE, para la obtención de espectroscopía infrarroja de estrellas gigantes rojas del bulbo, la barra, el disco y el halo.

SEGUE añade a la cobertura inicial del SDSS 15 bandas de longitud galáctica constante, dos de las cuales (las centradas en $l = 94^\circ$ y $l = 178^\circ$) llegan prácticamente al polo sur galáctico. En el presente Trabajo de Fin de Carrera hemos usado algunas de estas bandas para comparar los datos resultantes del modelo con datos de cuentas estelares en el visible.

5.2. Ajuste de los parámetros del modelo a los datos observacionales

Como se ha comentado, uno de los objetivos contemplados en la construcción del modelo es la posibilidad de ajustar los parámetros de la estructura galáctica a los datos observacionales, a fin de conseguir un mayor conocimiento de estos últimos.

Para este propósito primero se construyeron mapas de relevancia de cada componente estructural a fin de encontrar las zonas del cielo más apropiadas

para el ajuste de este componente. Posteriormente, se procedió al ajuste de los parámetros de algunos componentes a los datos de 2MASS.

5.2.1. Búsqueda de regiones para el ajuste de cada componente

Para el ajuste de los parámetros libres de cada componente estructural a los datos observacionales es preferible trabajar en una región del cielo en la cual las cuentas estelares aportadas por este componente sean relevantes en relación a las cuentas totales, puesto que en caso contrario se puede producir efectos perniciosos, como por ejemplo que la variación en las cuentas producida por la variación en los parámetros caiga dentro del margen del ruido poissoniano del total de cuentas, o que el algoritmo de ajuste quede atrapado en mínimos locales.

Por lo tanto, como primer paso antes de proceder al ajuste se realizaron mapas de relevancia de cada componente. A partir de estos mapas se determinaron las zonas del cielo en las cuales se realizarían los ajustes posteriores.

La implementación del modelo Galiac permite activar o desactivar la aportación de cada uno de los componentes estructurales. Para el cálculo del mapa de relevancia de un componente determinado, primeramente se calcula la predicción de las cuentas estelares en todo el cielo con todos los componentes activados, y posteriormente se procede a calcular la predicción de cuentas estelares activando tan sólo el componente en cuestión. Finalmente se construyen gráficamente los mapas que visualizan para cada coordenada galáctica (l, b) el porcentaje de contribución del componente i a las cuentas estelares totales, $relev(l, b) = 100 \cdot A_i(m, l, b) / A(m, l, b)$. En todos los cálculos se tiene en cuenta el efecto de la extinción.

En la figura 5.2 se muestran los mapas de relevancia obtenidos para el disco fino, el disco grueso, el bulbo, la barra, los brazos espirales y el halo. La integración de cuentas estelares se hizo para el filtro K hasta magnitud 14, y se han usado los parámetros por defecto indicados en el capítulo 3.

Podemos observar que la componente estructural de mayor peso es el disco fino, que en el plano puede llegar a suponer casi el 90 por ciento de las cuentas estelares totales y fuera del plano también mantiene una alta relevancia, con un promedio del 70 por ciento de las cuentas. En el intervalo encuadrado en $-20^\circ \leq l \leq 40^\circ$ y $-30^\circ \leq b \leq 30^\circ$ es la única región del cielo donde la preponderancia del disco fino cae significativamente (hasta niveles

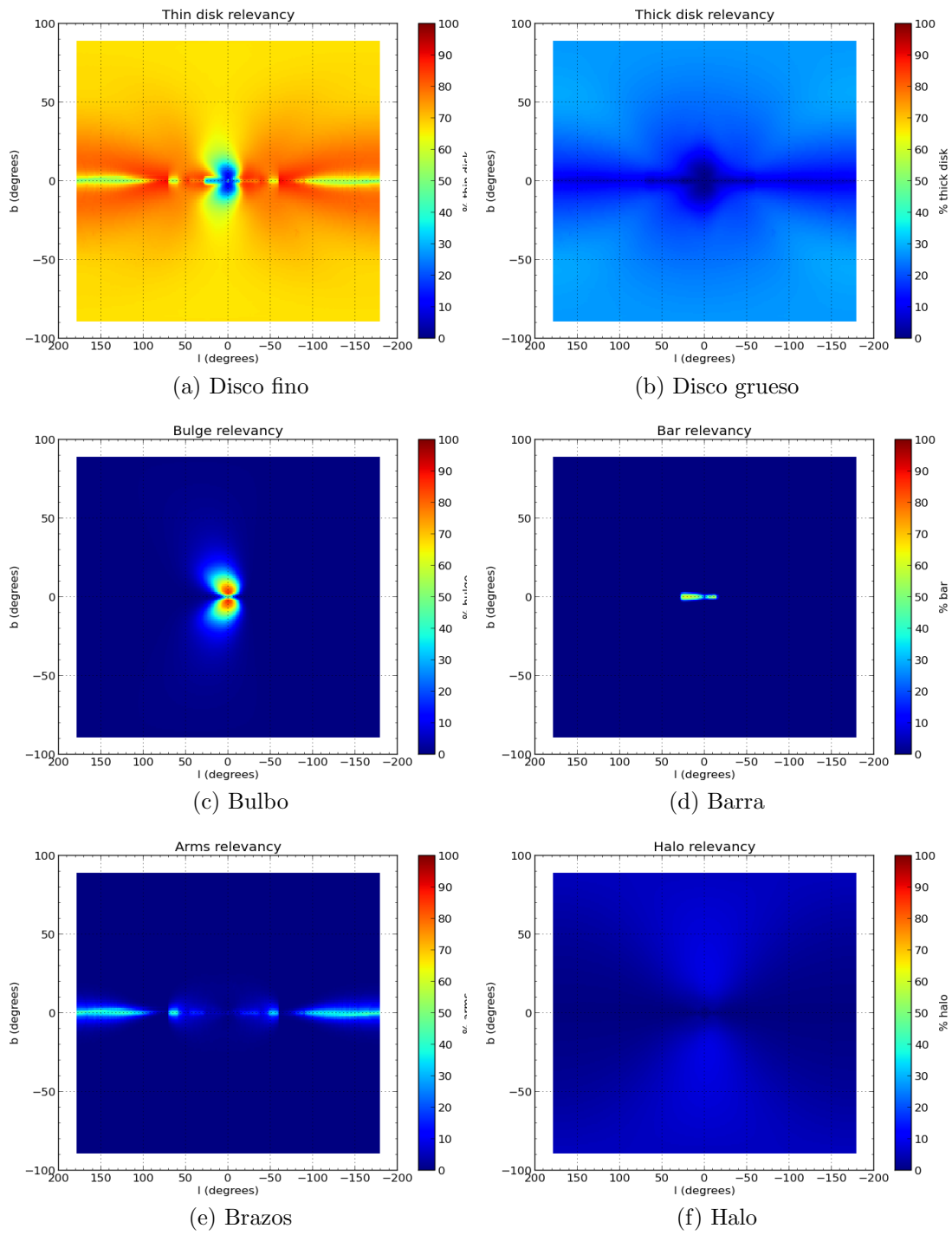


Figura 5.2: Relevancia de cada componente estructural de la Vía Láctea

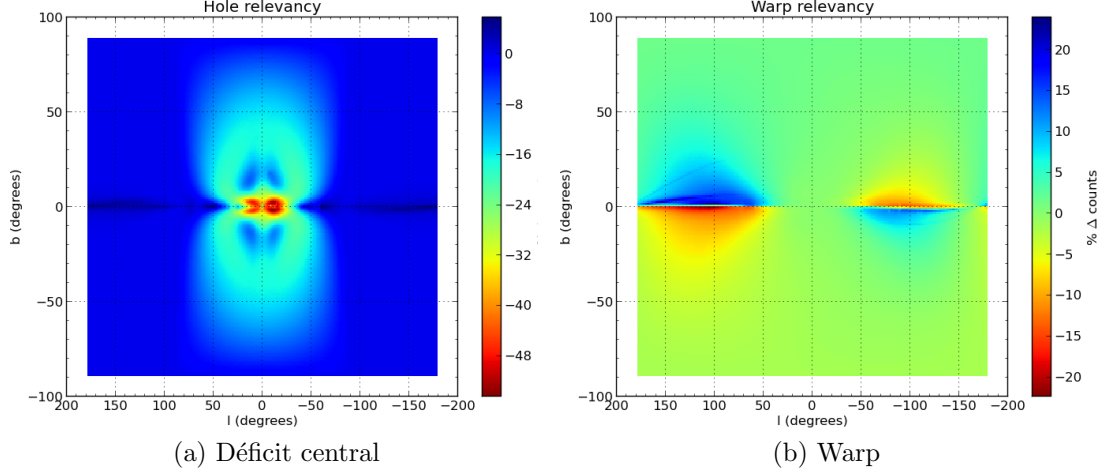


Figura 5.3: Variación en las cuentas estelares causada por las subestructuras del disco

muy por debajo del 30 %), puesto que es la que comprende las líneas de visión hacia el centro galáctico y, por lo tanto, esta caída se debe al efecto del déficit central de estrellas en el disco y a la mayor densidad estelar en los componentes centrales (bulbo y barra). Además, en una escala menor también se observa el efecto de los brazos en algunas regiones del plano, provocando una caída del porcentaje de relevancia del disco fino hasta el 50 % en algunas regiones situadas en $b \leq |2^\circ|$. También se deben evitar las zonas donde están presentes las nubes de Magellanes, por ser fuentes extragalácticas que no están modeladas en Galiac. Por lo tanto, la región escogida para hacer el ajuste de los parámetros del disco fino es

$$\begin{aligned}
 \mathcal{R}_{thind} = & (-180^\circ \leq l \leq 180^\circ, 30^\circ \leq |b| \leq 90^\circ) \\
 & \cup (-180^\circ \leq l \leq -20^\circ, 4^\circ \leq |b|) \\
 & \cup (40^\circ \leq l \leq 180^\circ, 4^\circ \leq |b|)
 \end{aligned} \tag{5.1}$$

El disco grueso tiene una relevancia mucho menor que el disco fino, pero bastante significativa en las regiones del cielo alejadas del plano y del centro galáctico, donde alcanza relevancias del 25 al 30 por ciento de las cuentas (y por lo tanto, en estas regiones la contribución del disco fino más la del disco grueso supone valores entorno al 95 % de las cuentas totales). La región de

cielo escogida para el ajuste de los parámetros del disco grueso es

$$\mathcal{R}_{thickd} = (-180^\circ \leq l \leq 180^\circ, 40^\circ \leq |b| \leq 90^\circ) \quad (5.2)$$

El bulbo es una componente con una gran densidad estelar, y por lo tanto las zonas del cielo cuyas líneas de visión atraviesan el bulbo están dominadas por esta componente. En el mapa de relevancia de la figura 5.2c se puede observar que hay regiones donde el bulbo alcanza valores superiores al 80 % de las cuentas totales. Se observa que la región en que domina el bulbo no es simétrica respecto al eje $l = 0^\circ$, a causa de la triaxialidad del bulbo y del ángulo que presenta respecto a la línea Sol-centro. Además, en el plano galáctico se nota el efecto de la presencia de la barra y también el efecto de la alta extinción en esta zona, que causa que sólo sean observables las cuentas estelares procedentes del disco (lo cual es el responsable de la forma lobulada que se observa en el gráfico en $0 \leq l \leq 40^\circ$). Teniendo en cuenta esta distribución, la zona escogida para realizar los ajustes del bulbo sería

$$\mathcal{R}_{bulge} = (-15^\circ \leq l \leq 25^\circ, 2^\circ \leq |b| \leq 20^\circ) \quad (5.3)$$

La barra aporta entre un 50 y 60 por ciento de las cuentas estelares totales en una región muy localizada, que se extiende en longitudes del rango -15° y 30° , y en una banda muy estrecha de latitudes entre $-1,5^\circ$ y $2,5^\circ$,

$$\mathcal{R}_{bar} = (-15^\circ \leq l \leq 30^\circ, -1,5^\circ \leq b \leq 2,5^\circ) \quad (5.4)$$

Los brazos están localizados en algunas regiones del plano galáctico, concretamente en aquellas longitudes galácticas en las cuales la línea de visión es tangencial a la espiral. Estas regiones se extienden por todo el rango de longitudes, pero en cambio en un rango estrecho de latitudes,

$$\mathcal{R}_{arms} = (-180^\circ \leq l \leq 180^\circ, -5^\circ \leq b \leq 5^\circ) \quad (5.5)$$

El halo es la componente que menos aporta a las cuentas estelares totales, menos de un 8 %. Para hacer el ajuste de esta componente estructural conviene tener en cuenta las zonas del cielo con menos contribución del disco, yendo hacia las regiones de los polos galácticos. Aparte de las zonas de los polos, en el mapa de relevancia también se observan dos regiones de menor latitud en las cuales el porcentaje de cuentas aportadas por el halo alcanzan

su máximo, en $|l| \leq 30^\circ$ y $20^\circ \leq |b| \leq 70^\circ$. Por lo tanto, la zona apropiada para el ajuste del halo es

$$\begin{aligned} \mathcal{R}_{halo} = & (|l| \leq 100^\circ, 50^\circ \leq |b|) \\ & \cup (|l| \leq 30^\circ, 20^\circ \leq |b| \leq 70^\circ) \end{aligned} \quad (5.6)$$

Finalmente, queremos realizar también ajustes de los parámetros de las subestructuras del disco, como es el caso del alabeo y del déficit central. En la figura 5.3 se puede localizar las zonas del cielo en las cuales la contribución de estas subestructuras es importante.

5.2.2. Resultado del ajuste a los datos observacionales

Para realizar los ajustes de los parámetros libres del modelo se usaron los datos observacionales del catálogo de fuentes puntuales (Point Source Catalog) de 2MASS. En concreto, se ha trabajado con los datos de la banda K y hasta magnitud 13. El método que se ha seguido es el detallado en la sección 3.5

En la presente sección se detallan los resultados obtenidos con los ajustes.

Ajuste de los parámetros del disco fino

El primer paso que hemos realizado ha sido el ajuste de los parámetros estructurales del disco fino en la zona externa del mismo. En concreto, los parámetros que hemos ajustado son la longitud de escala, $h_{r\{f\}}$, la escala de altura en R_\odot , $h_{z\{f\}}(R_\odot)$, la longitud de escala del ensanchamiento, $h_{R,flare}$ y el radio galactocéntrico en el cual empieza el mismo, R_{flare} . El resto de los parámetros del modelo se han dejado a su valor por defecto, exceptuando que se ha desactivado el alabeo y los brazos espirales. La zona que se ha elegido es $(|l| > 50, 5 < |b| < 50)$, a fin de quitar las partes más internas del disco, las que se encuentran demasiado cercanas al plano y las que empiezan a tener una contribución apreciable del disco grueso.

El ajuste se ha realizado sobre 10553 puntos en la región seleccionada. Se han necesitado 30 iteraciones hasta la convergencia, lo cual ha supuesto un tiempo de cálculo de 203,5 segundos en el cluster MPI de tres nodos.

En la tabla 5.1 se detallan los resultados del ajuste. Podemos observar que el radio galactocéntrico en el cual empieza el ensanchamiento ha variado notablemente respecto a su valor por defecto, hacia a una posición cercana al

radio solar, las longitudes de escala del disco y del ensanchamiento también han variado, creciendo ligeramente, y el resto de parámetros han recibido poca variación. En la figura 5.4 se pueden ver gráficamente los resultados del ajuste.

El hecho de que salgan parámetros ligeramente distintos a los valores por defecto de López-Corredoira et al. (2002) puede deberse a que aquellos usaban 2MASS hasta magnitud 14 en K en algunas regiones, mientras que aquí utilizamos todo el espacio R_{thind} y hasta magnitud 13 en K. Errores en la función de luminosidad promedio es probable que expliquen estas diferencias.

Parámetro	Valor
$\rho_{\odot\{f\}}$	$54,08 \times 10^6 \star \cdot \text{kpc}^{-3}$
$h_{r\{f\}}$	2,52 kpc
$h_{z\{f\}}(R_{\odot})$	0,257 kpc
$h_{R,flare}$	6,76 kpc
R_{flare}	9,27 kpc

Tabla 5.1: Resultado del primer ajuste del disco fino

Ajuste de los parámetros del disco grueso

Una vez obtenidos la longitud de escala y la escala de altura del disco fino, procedemos a ajustar los parámetros del disco grueso. Usamos la zona definida en 5.2, cambiamos los parámetros del disco fino por los de obtenidos en el apartado anterior, y procedemos a ajustar $\rho_{\odot\{g\}}$, $h_{r\{g\}}$ y $h_{z\{g\}}(R_{\odot})$. Los resultados del ajuste se listan en la tabla 5.2.

Parámetro	Valor
$\rho_{\odot\{g\}}$	$4,99 \times 10^6 \star \cdot \text{kpc}^{-3}$
$h_{r\{g\}}$	3,38 kpc
$h_{z\{g\}}(R_{\odot})$	0,643 kpc

Tabla 5.2: Resultado del primer ajuste del disco grueso

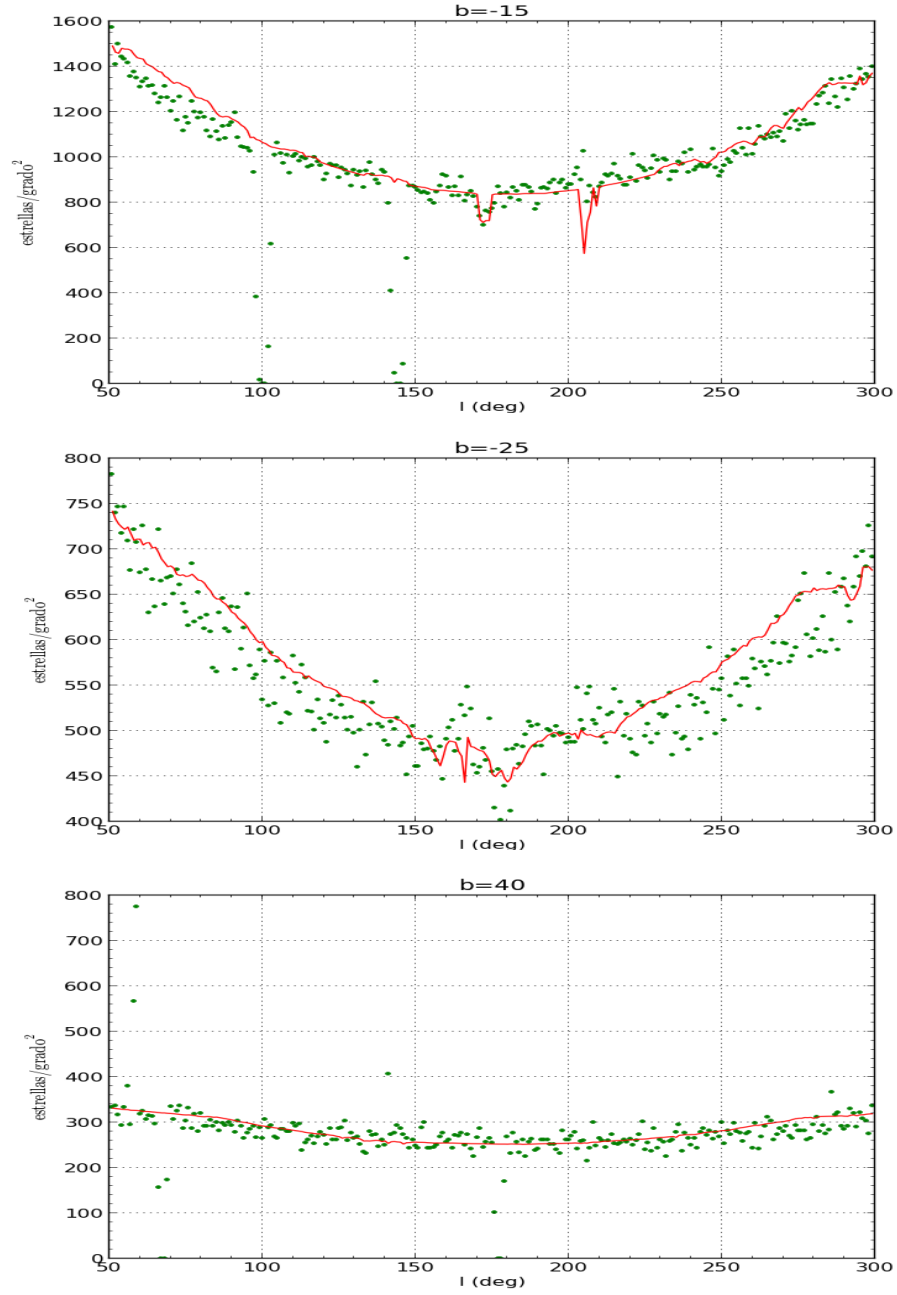


Figura 5.4: Ajuste del disco fino externo

Ajuste del bulbo y la parte central del disco

Para realizar el ajuste de los parámetros del bulbo se intentaron inicialmente varias aproximaciones en las cuales se ajustaban iterativamente los diferentes parámetros por separado, pero los mejores resultados se han obtenido ajustando todos los parámetros simultáneamente. Además, el ajuste también mejoró cuando se incluyeron los parámetros del déficit central de estrellas del disco, puesto que esta característica tiene un gran impacto en las cuentas estelares de la zona de cielo óptima que se determinó en (5.3) para el ajuste del disco.

El ajuste se ha realizado sobre un total de 1404 puntos. Ha requerido un total de 49 iteraciones, que han supuesto 25 segundos de cálculo en el clúster MPI de tres nodos.

En la tabla 5.3 se detallan los resultados del ajuste. En la figura 5.5 se puede ver la comparación de las cuentas estelares con 2MASS configurando los nuevos parámetros en el modelo.

Ajuste de los parámetros de los brazos espirales

A continuación hemos seguido con el ajuste de los parámetros de los brazos espirales. Inicialmente habíamos optado por hacer el ajuste de los brazos y del alabeo simultáneamente, puesto que coinciden en zonas amplias, pero el resultado fue peor que realizando el ajuste por separado.

En la tabla 5.4 se listan los parámetros resultantes del ajuste. En la figura 5.6 se puede comparar el efecto de activar los brazos en el modelo con los nuevos parámetros encontrados.

Parámetro	Valor
α_{bulge}	$27,98 \pm 1,27^\circ$
$\rho_{bulge}(0)$	$(1,05 \pm 0,03) \times 10^{10} \star \cdot \text{kpc}^{-3}$
h_b	$0,538 \pm 0,013 \text{ kpc}$
r_y	$0,40 \pm 0,01$
r_z	$0,38 \pm 0,01$
$H_{R,hole}$	$0,54 \pm 0,06 \text{ kpc}$
$C_{h,1}$	$0,203 \pm 0,013 \text{ kpc}^{-1}$
$C_{h,2}$	$0,031 \pm 0,002 \text{ kpc}^{-2}$

Tabla 5.3: Ajuste de los parámetros del bulbo y del déficit central del disco

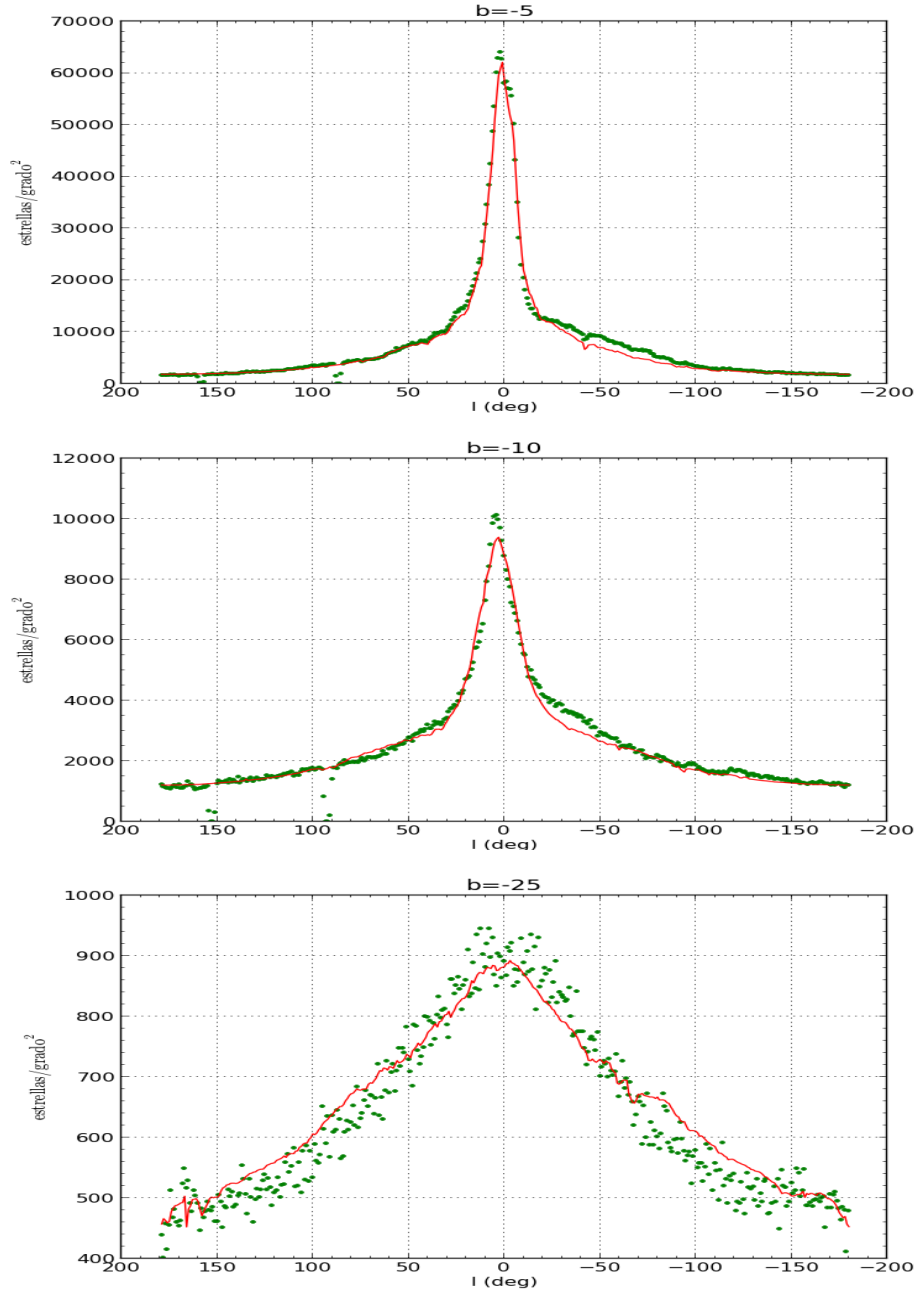


Figura 5.5: Ajuste del bulbo y el déficit central del disco

Parámetro	Valor
f_{arm}	0,08
α_0	260°
r_0	1,4 kpc
$\arctan b$	$8,34^\circ$
σ_{arm}	0,104 kpc
$h_{z\{arm\}}(R_\odot)$	0,09 kpc

Tabla 5.4: Resultado del ajuste de los parámetros de los brazos espirales

Podemos ver que, a diferencia de los casos anteriores, el efecto del ajuste no es tan notable. Se mejora bastante algunas zonas, como la de $(-20^\circ < l < -100^\circ, |b| < 10^\circ)$, pero empeoran ligeramente otras. En algunos de los puntos tangentes de los brazos de la tabla 3.5 se aprecia mejora, pero en otros no. La conclusión que obtenemos es que es difícil realizar un buen ajuste de los brazos con los datos disponibles en la actualidad, lo cual se refleja también en el poco consenso que hay sobre el tema en la literatura.

Ajuste de los parámetros de la barra

Finalmente, procedemos a realizar el ajuste de la barra larga. Para ello, usamos la zona definida en (5.4). Los resultados se listan en la tabla 5.5, en los cuales podemos ver que el ajuste ha dejado la longitud de la barra en un valor similar al de partida, pero que en cambio la altura de la barra se ha visto reducida a la mitad. Esto podría ser debido a que en este ajuste se tenían en cuenta también los brazos espirales. Con los brazos espirales y con los parámetros originales de la barra hay un exceso de cuentas en la región $1 < |b| < 3$. Esto último también puede estar causado parcialmente por el hecho de que en el catálogo estén infravaloradas las cuentas estelares en esta zona, debido a la confusión de fuentes.

En la figura 5.7 puede verse el resultado del ajuste.

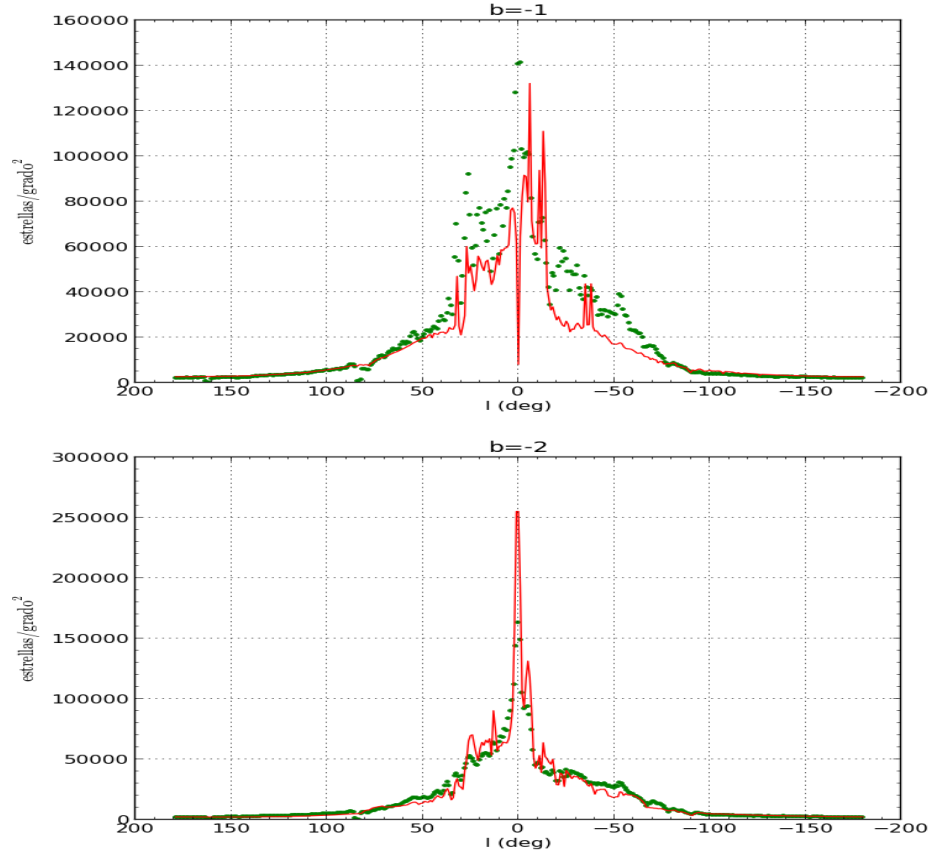


Figura 5.6: Ajuste de los brazos espirales

Parámetro	Valor
α_{bar}	$42,49^\circ$
$\rho_{bar}(0)$	$4,64 \times 10^9 \star \cdot \text{kpc}^{-3}$
len_{bar}	$3,7\text{kpc}$
$\sigma_{y,bar}$	$0,51\text{kpc}$
$\sigma_{z,bar}$	$0,074\text{kpc}$

Tabla 5.5: Resultado del ajuste de la barra

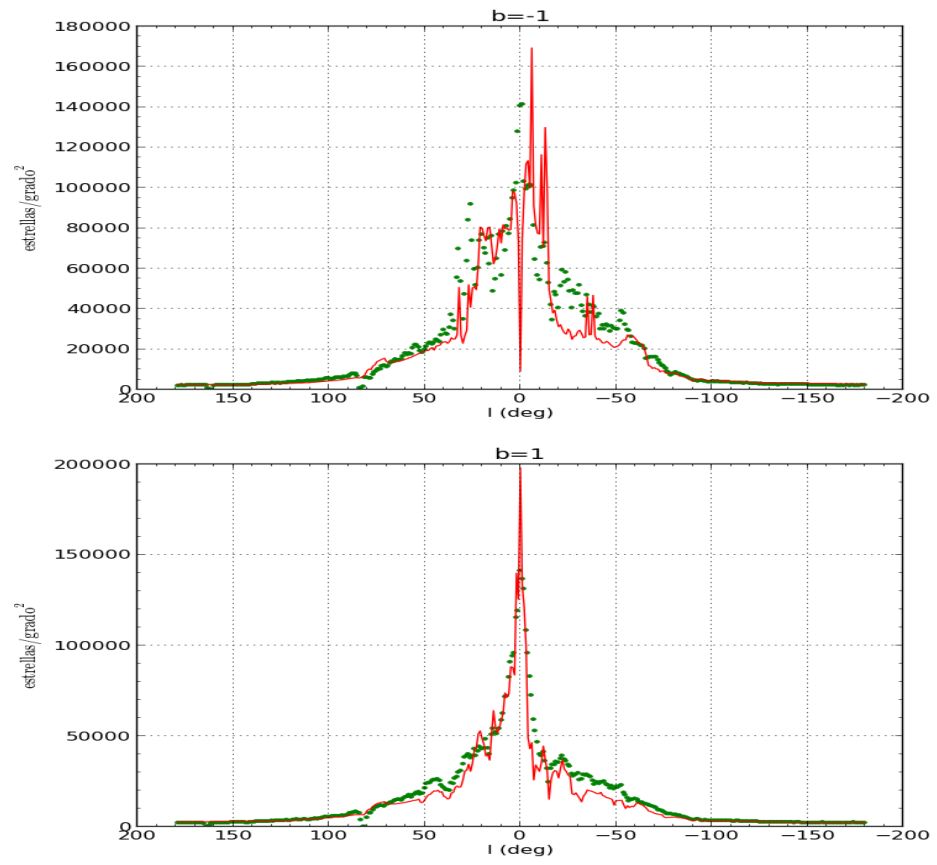


Figura 5.7: Ajuste de la barra

5.3. Comparación multibanda con datos observacionales

Para concluir este capítulo compararemos el resultado del modelo con datos procedentes de cartografiados en diferentes longitudes de onda.

En la figura 5.8 podemos ver el resultado de aplicar el modelo en varias regiones del cielo en la banda K hasta magnitud 13, y en la figura 5.9 podemos ver las mismas regiones del cielo, pero en la banda J y también hasta magnitud 13.

Finalmente, en la figura 5.10 podemos ver la predicción del modelo para la banda V hasta magnitud 20, y se comparan con datos de una de las tiras de longitud galáctica constante de Sloan ($l = 110^\circ$). Los datos de Sloan corresponden al filtro g, que se ha transformado a magnitudes en V del sistema de Johnson mediante las expresiones de Jester et al. (2005).

Podemos ver que en bandas diferentes de K, aunque el modelo sigue aproximadamente las cuentas observacionales, existen algunas desviaciones. Ello está causado, con mucha probabilidad, por el hecho de que puedan haber incorrecciones en las funciones de luminosidad. En K el resultado es mejor puesto que hemos usado esta banda para los ajustes. Posiblemente, la concordancia del modelo en varias bandas se puede mejorar si se realizaran los ajustes con datos de múltiples longitudes de onda.

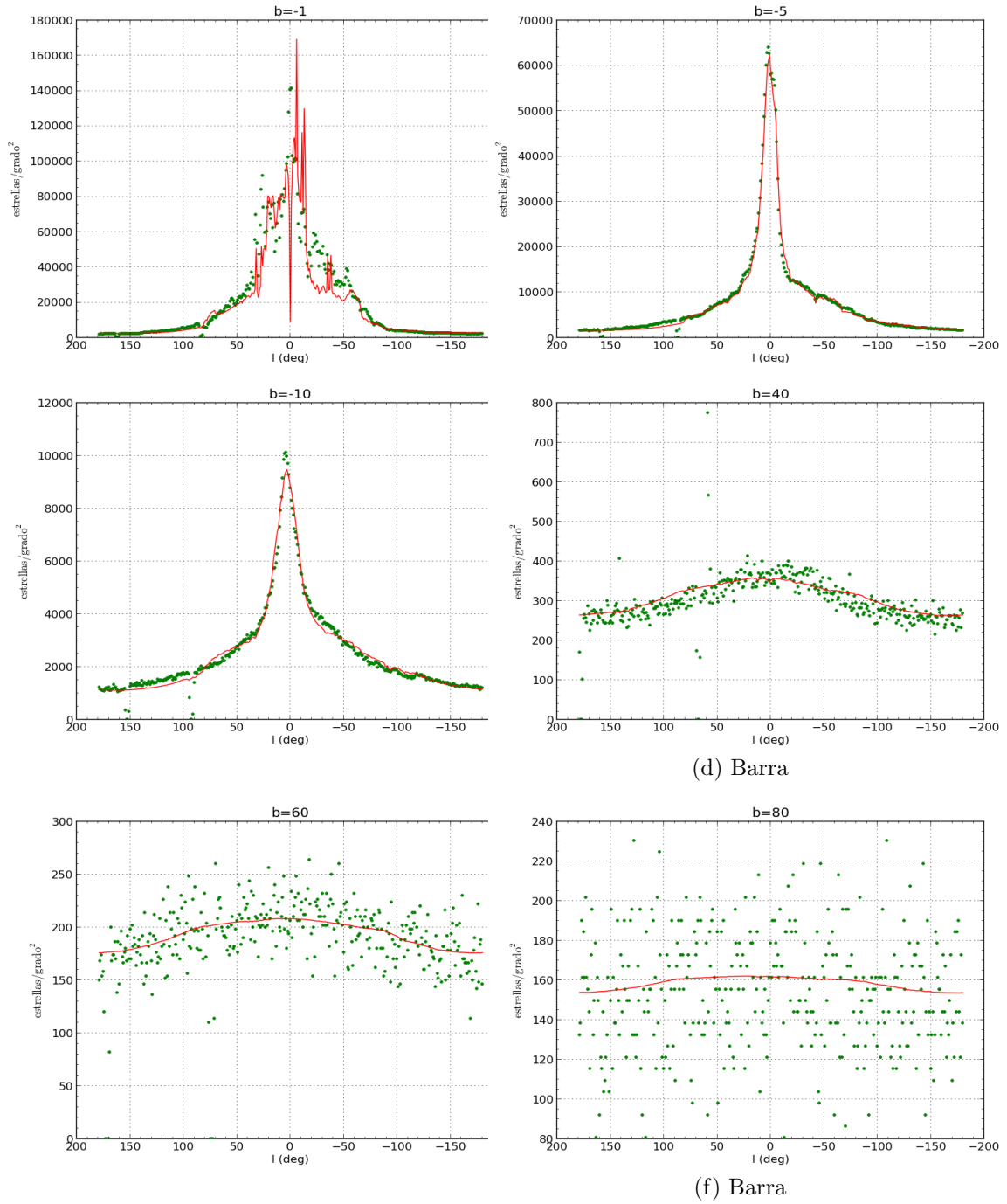


Figura 5.8: Comparación del resultado del modelo en la banda K (mag=13) con datos de 2MASS

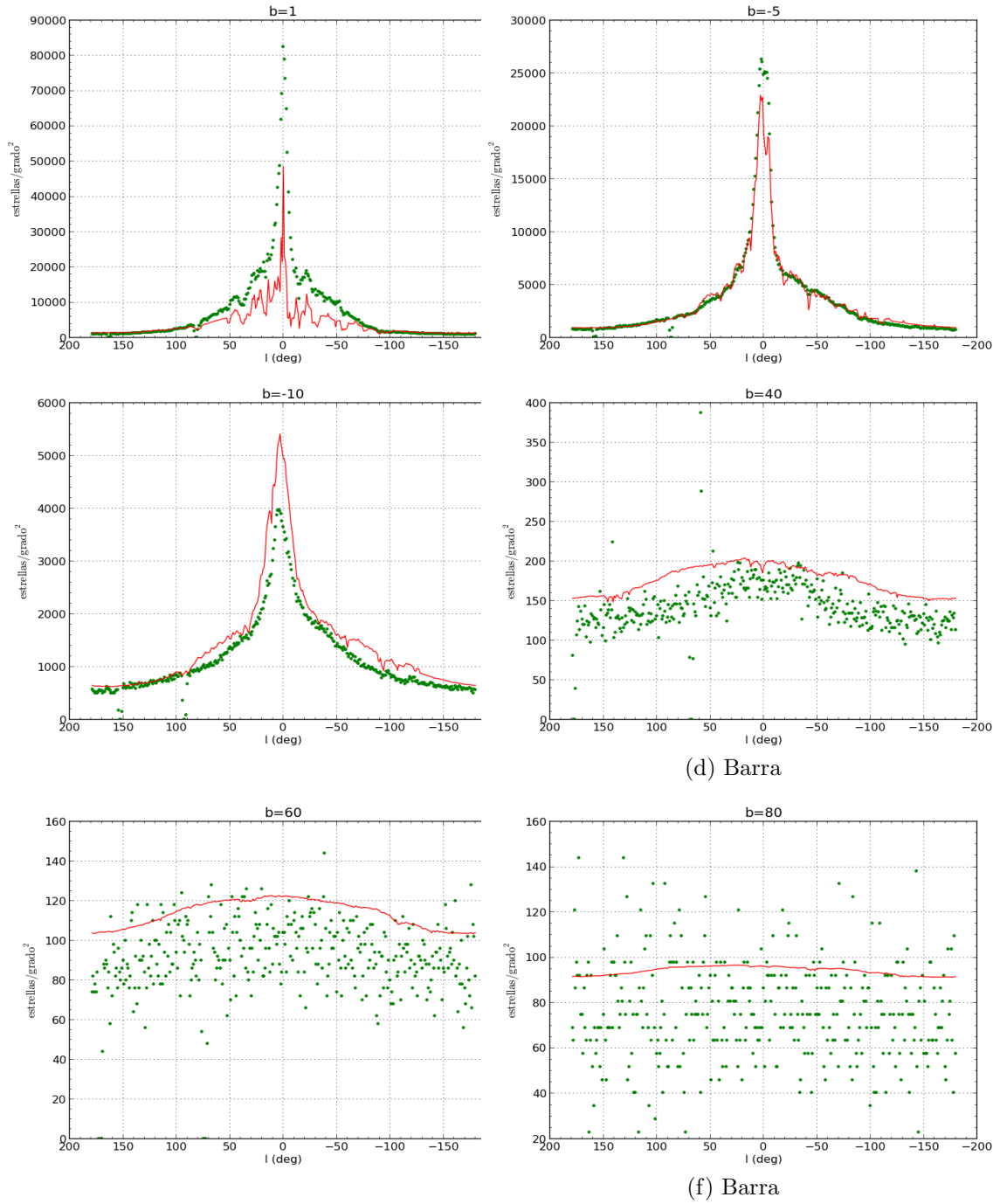


Figura 5.9: Comparación del resultado del modelo en la banda J (mag=13) con datos de 2MASS

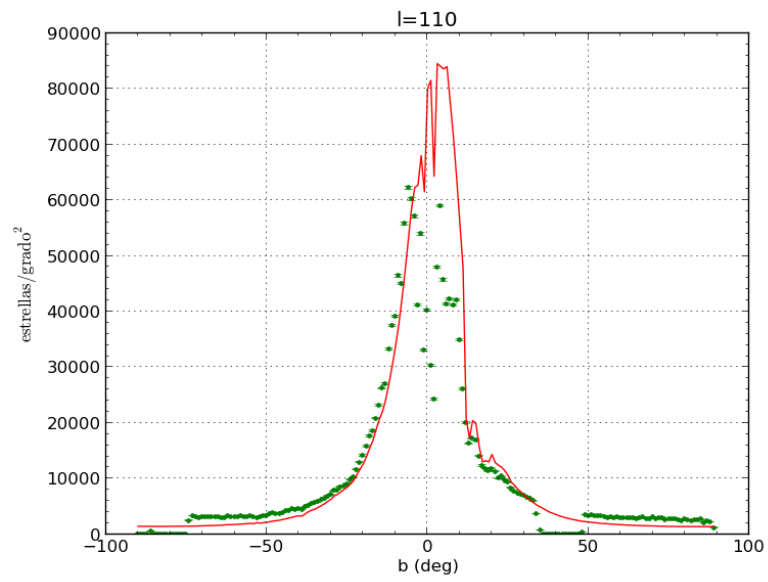


Figura 5.10: Comparación del resultado del modelo en la banda V (mag=20) con datos de Sloan

Capítulo 6

Conclusión

En el contexto del presente Trabajo de Fin de Carrera se ha desarrollado el código informático de un modelo de cuentas estelares galáctico, y posteriormente se ha usado para realizar ajustes de los parámetros estructurales de los componentes de la Vía Láctea a los datos observacionales procedentes disponibles. Este trabajo ha servido como introducción al alumno al campo de la investigación de la morfología galáctica, pero también ha generado una herramienta que se pondrá a disposición de la comunidad y se espera que pueda servir como base de futuras investigaciones.

Hay varios puntos que creemos apropiado destacar del modelo que se ha implementado. Uno de ellos es el hecho de haber usado un mapa detallado de extinción. Esto permite obtener resultados de cuentas estelares más realistas (como se aprecia en la figura 5.1), puesto que las variaciones a pequeña escala debidas a la presencia de nubes de polvo (que no son modelables) quedan reflejadas correctamente.

Por otra parte, otra de las ventajas del modelo es el hecho que tiene en cuenta todos los componentes estructurales de gran escala que, según el conocimiento actual, forman parte de la Vía Láctea, y además está actualizado a los últimos resultados de la literatura. Gracias a ello, se puede usar este modelo como punto de partida para nuevas investigaciones. Adicionalmente, puesto que el modelo incluye varias longitudes de onda, es posible usarlo en comparación a una variedad mayor de catálogos estelares (o incluso extenderlo a futuros sistemas fotométricos).

Otra particularidad a destacar es la posibilidad de ejecutar el modelo en sistemas de computación paralelos, de forma que el tiempo total de cálculo se reduce muy considerablemente. Además, se ha dotado al modelo de varias

interfaces de usuario, con lo cual se responde a necesidades diferentes según el perfil de los investigadores que lo usen.

6.1. Trabajos futuros

El autor no considera el modelo como un trabajo ya cerrado, sino que existe la intención de seguir avanzando en él. Por una parte, mientras se estaba realizando el desarrollo y el estudio de los resultados han surgido varias ideas de mejora; por otra parte, ya desde el inicio del proyecto existía la idea de ir extendiendo el modelo en trabajos futuros, con el objetivo último de un modelo autoconsistente.

Una de las ideas de mejora que han surgido está relacionada con el mapa de extinción. El mapa tridimensional de Drimmel et al. (2003) está basado en un modelo de la distribución del polvo que difiere de la que nosotros usamos para la distribución estelar. Tomando como hipótesis que la distribución del polvo debe mimetizar la distribución de las estrellas, una posible mejora sería repetir el trabajo de Drimmel pero ajustando sus datos a nuestro modelo.

También se puede mejorar el modelo introduciendo funciones de luminosidad más actualizadas para todos los filtros, o incluso experimentando con funciones de luminosidad sintéticas.

Otra tarea a la que se quiere dedicar más tiempo es el ajuste de los datos. Aunque en el presente trabajo ya se han realizado algunos ajustes como prueba de concepto del modelo, con mucha seguridad será posible refinar los resultados. Por ejemplo, se podría probar a realizar los ajustes en una banda diferente a K.

Finalmente, el modelo actual es un punto de partida hacia un modelo más complejo. Un objetivo de interés para el grupo de investigación de morfología galáctica es el de conseguir un modelo galáctico autoconsistente, en el cual no sólo las cuentas estelares sean coherentes con las observaciones, sino que también lo sean el potencial y la cinemática que se deduzcan de las funciones de densidad y unas condiciones iniciales. Entre otros aspectos, sería necesario que el modelo proporcionase información sobre masas y velocidades, cosa que actualmente no tiene disponible. Un posible camino sería la implementación de funciones de luminosidad y funciones iniciales de masa sintéticas: se especificarían como datos iniciales los rangos de metalicidades e historias de formación estelar de cada uno de los componentes, y mediante librerías de evolución estelar se obtendrían los parámetros necesarios. Poste-

riormente se dejaría evolucionar el conjunto, con las restricciones aportadas por las ecuaciones de Poisson y de Boltzman.

Capítulo 7

Bibliografía

- J. N. Bahcall. Star counts and galactic structure. *Annual review of astronomy and astrophysics*, 24:577–611, 1986.
- J. N. Bahcall and R. M. Soneira. The universe at faint magnitudes. i - models for the galaxy and the predicted star counts. *Astrophysical Journal Supplement Series*, 44:73–110, 1980.
- F. N. Bash. Does the galaxy have four spiral arms. *Astrophysical Journal*, 250:551–560, 1981.
- D. Becker, T. Sterling, D. Savarese, J. Dorband, U. Ranawake, and C. Packer. Beowulf: A parallel workstation for scientific computation. *International Conference on Parallel Processing (Proceedings)*, 95, 1995.
- R. A. Benjamin, E. Churchwell, B. L. Babler, R. Indebetow, and M. R. Meade. First glimpse results on the stellar structure of the galaxy. *Astrophysical Journal*, 630:L149–L152, 2005.
- J. Binney and M. Merrifield. *Galactic astronomy*. Princeton University Press, 1998.
- J. Bochansky, S. Hawley, K. Covey, A. West, N. Reid, D. Golimowsky, and Z. Ivezić. The luminosity and mass functions of low-mass stars in the galactic disk. ii. the field. *The Astronomical Journal*, 139(6):2679–2699, 2010.

- R. Buser, J. Rong, and S. Karaali. The new basel high-latitude field star survey of the galaxy. ii. the thick disk component: density structure, luminosity function, and metallicity distribution. *Astronomy and Astrophysics*, 348:98–112, 1999.
- A. Cabrera-Lavers. *Análisis morfológico multibanda del contenido estelar del plano y disco de la Vía Láctea*. PhD thesis, Universidad de la Laguna, 2004.
- R. Drimmel and D. Spergel. Three-dimensional structure of the milky way disk: The distribution of stars and dust beyond $0,35r_{\odot}$. *The Astrophysical Journal*, 556:181–202, 2001.
- R. Drimmel, Antonio Cabrera-Lavers, and Martín López-Corredoira. A three-dimensional galactic extinction model. *Astronomy and Astrophysics*, 409:205–215, 2003.
- N. Eaton, D. J. Adams, and A. B. Giles. The 2.2 micron stellar distribution in the galactic plane. *Monthly Notices of the Royal Astronomical Society*, 208:241–251, 1984.
- P. Englmaier and O. Gerhard. Gas dynamics and large-scale morphology of the milky way galaxy. *Monthly Notices of the Royal Astronomical Society*, 304:512–534, 1999.
- C. Francis and E. Anderson. Evidence of a bisymmetric spiral in the milky way. *Monthly Notices of the Royal Astronomical Society*, 422(2):1283–1293, 2012.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Number ISBN 0-201-63361-2. Addison-Wesley, 1994.
- F. Garzón, P. L. Hammersley, T. Mahoney, X. Calbet, M. J. Selby, and I. D. Hepburn. A two-micron galactic survey. *Monthly Notices of the Royal Astronomical Society*, 264(3), 1993.
- Y. M. Georgelin and Y. P. Georgelin. The spiral structure of our galaxy determined from h ii regions. *Astronomy and Astrophysics*, 49:57–79, 1976.

- G. Gilmore and N. Reid. New light on faint stars. iii - galactic structure towards the south pole and the galactic thick disc. *Monthly Notices of the Royal Astronomical Society*, 202:1025–1047, 1983.
- L. Girardi, M. A. T. Groenewegen, E. Hatziminaoglou, and L. da Costa. Star counts in the galaxy. simulating from very deep to very shallow photometric surveys with the trilegal code. *Astronomy and Astrophysics*, 436(3):895–915, 2005.
- P. L. Hammersley, F. Garzon, T. Mahoney, and X. Calbet. Infrared signatures of the inner spiral arms and bar. *Monthly Notices of the Royal Astronomical Society*, 269, 1994.
- P. L. Hammersley, F. Garzon, T. Mahoney, and X. Calbet. The tilted old galactic disc and the position of the sun. *Monthly Notices of the Royal Astronomical Society*, 273, 1995.
- J. Holtzman, A. Watson, W. Baum, C. Grillmair, E. Groth, R. Light, R. Lynds, and E. O’Neil. The luminosity function and initial mass function in the galactic bulge. *The Astronomical Journal*, 115(5):1946–1957, 1989.
- S. Jester, D. Schneider, G. Richards, R. Green, M. Schmidt, P. Hall, M. Strauss, D. Vanden Berk, C. Stoughton, J. Gunn, and J. Brinkmann. The sloan digital sky survey view of the palomar-green bright quasar survey. *The Astronomical Journal*, 130(3):873–895, 2005.
- P. Kruchten. Architectural blueprints - the “4+1” view model of software architecture. *IEEE Software*, 12(6):42–50, 1995.
- C. Larman. *Applying UML and Patterns – An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Number ISBN 0-13-148906-2. Prentice Hall, 2005.
- E. S. Levine, L. Blitz, and C. Heiles. The vertical structure of the outer milky way hi disk. *The Astrophysical Journal*, 643(2):881–896, 2006a.
- E. S. Levine, L. Blitz, and C. Heiles. The spiral structure of the outer milky way in hydrogen. *Science*, 312(5781):1773–1777, 2006b.
- C. C. Lin and F. Shu. On the spiral structure of disk galaxies, ii. outline of a theory of density waves. *Proceedings of the National Academy of Sciences of the United States of America*, 55(2):229–234, 1966.

- M. López-Corredoira, F. Garzón, P. Hammersley, T. Mahoney, and X. Calbet. The morphology and luminosity function of the galactic bulge from tmgs star counts. *Monthly Notices of the Royal Astronomical Society*, 292:L15–L19, 1997.
- M. López-Corredoira, F. Garzón, J. E. Beckman, T. Mahoney, P. Hammersley, and X. Calbet. A major star formation region in the receding tip of the stellar galactic bar. ii. supplementary information and evidence that the bar is not the same structure as the triaxial bulge previously reported. *The Astronomical Journal*, 118:381–389, 1999.
- M. López-Corredoira, A. Cabrera-Lavers, F. Garzón, and P. Hammersley. Old stellar galactic disc in near-plane regions according to 2mass: Scales, cut-off, flare and warp. *Astronomy and Astrophysics*, 394:883–899, 2002.
- M. López-Corredoira, A. Cabrera-Lavers, O. Gerhard, and F. Garzón. Evidence for a deficit of young and old stars in the milky way inner in-plane disc. *Astronomy and Astrophysics*, 421:953–967, 2004.
- M. López-Corredoira, A. Cabrera-Lavers, and O. Gerhard. A boxy bulge in the milky way. inversion of the stellar statistics equation with 2mass data. *Astronomy and Astrophysics*, 439:107–110, 2005.
- M. López-Corredoira, A. Cabrera-Lavers, T. Mahoney, P. Hammersley, F. Garzón, and C. González-Fernández. The long bar in the milky way: corroboration of an old hypothesis. *The Astronomical Journal*, 133:154–161, 2007.
- D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11 (2):431–441, 1963.
- P. G. Mezger, R. Zylka, S. Philipp, and R. Launhardt. The nuclear bulge of the galaxy. ii. the k band luminosity function of the central 30 pc. *Astronomy and Astrophysics*, 348:457–465, 1999.
- M. Mueller and W. Arnett. Propagating star formation and irregular structure in spiral galaxies. *The Astrophysical Journal*, 210:670–678, 1976.
- G. Neugebauer and R. B. Leighton. Two-micron sky survey. a preliminary catalogue. *NASA SP*, 1969.

- N. Reid, J. Gizis, and S. Hawley. The palomar/msu nearby star spectroscopic survey. iv. the luminosity function in the solar neighborhood and m dwarf kinematics. *The Astronomical Journal*, 124(5):2721–2738, 2002.
- C. Reyl   and A. C. Robin. Early galaxy evolution from deep wide field star counts. ii. first estimate of the thick disc mass function. *Astronomy and Astrophysics*, 373:886–894, 2001.
- G. Rieke and M. Lebofsky. The interstellar extinction law from 1 to 13 microns. *The Astrophysical Journal*, 288:618:621, 1985.
- A. C. Robin, C. Reyl  , S. Derri  re, and S. Picaud. A synthetic view on structure and evolution of the milky way. *Astronomy and Astrophysics*, page 405:523, 2003.
- D. H. Rogstad, I. A. Lockhart, and M. C. H. Wright. Aperture-synthesis observations of h i in the galaxy m83. *The Astrophysical Journal*, 193:309:319, 1974.
- P. Rojo and M. Ruiz. Kinematics and luminosity function of dwarf populations in three areas of the cal  n-eso proper-motion catalog. *The Astronomical Journal*, 126:353–369, 2003.
- M. Skrutskie, R.M. Cutri, R. Stiening, M.D. Weinberg, S. Schneider, J.M. Carpenter, C. Beichman, R. Capps, T. Chester, J. Elias, J. Huchra, J. Liebert, C. Lonsdale, D.G. Monet, S. Price, P. Seitzer, T. Jarrett, J.D. Kirkpatrick, J. Gizis, E. Howard, T. Evans, J. Fowler, L. Fullmer, R. Hurt, R. Light, E.L. Kopan, K.A. Marsh, H.L. McCallon, R. Tam, S. Van Dyk, and S. Wheelock. The two micron all sky survey (2mass). *The Astronomical Journal*, 131(2):1163–1183, 2006.
- J. H. Taylor and J. M. Cordes. Pulsar distances and the galactic distribution of free electrons. *The Astrophysical Journal*, 411(2):674–684, 1993.
- von Seeliger. *Abh. Bayerische Akad. Wiss. Math.-Phys. Kl*, 1898.
- R. Wainscoat, M. Cohen, K. Volk, H. Walker, and D. Schwartz. A model of the 8-25 micron point source infrared sky. *Astrophysical Journal Supplement Series*, 83(1):111:146, 1992.

- R. White and S. Stemwedel. The quadrilateralized spherical cube and quad-tree for all sky data. *Astronomical Data Analysis Software and Systems I, A.S.P. Conference Series*, 25:379, 1992.
- R. Wielen. The kinematics and ages of stars in gliese’s catalogue. *Highlights of astronomy*, 3:395–407, 1974.
- Z. Zheng, C. Flynn, A. Gould, J. Bahcall, and S. Salim. M dwarfs from hubble space telescope star counts. v. the i-band luminosity function. *The Astrophysical Journal*, 601(1):500–501, 2004.